NIST GCR 94-658

SOFTWARE FOR MANUFACTURING

OPPORTUNITIES FOR INNOVATION

U.S. Department of Commerce
Technology Administration • National Institute of Standards and Technology

---

National Institute of Standards and Technology

Arati Prabhakar, *Director*

Technology Services

Peter L.M. Heydemann, *Director*

David E. Edgerly, *Deputy Director*

Technology Services

Series Editor

NIST GCR 94-658

Opportunities for Innovation: Software for Manufacturing

Edited by

David E. Edgerly
National Institute of Standards and Technology
Gaithersburg, MD 20899

Prepared for

U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899

Grant 50SBNB3C7626

August 1994

CONTENTS

Chapter 1
Approach

Dennis J. Donohue
President
SMS Software Management Incorporated
750 The City Drive, Suite 420
Orange, CA 92668
714/740-1113

Brian K. Seitz
B.K.Seitz & Associates
12555 Euclid Street, Unit 53
Garden Grove, California 92640
714/534-1258

Abstract

The purpose of this Monograph is to survey the existing software applicable to the Manufacturing Enterprises, to identify the gaps (see definition) and the opportunities for software developers to close these gaps. Information Technology is constantly growing and adapting, this Monograph proposes directions for this growth and is a snapshot of current conditions, to be used as a starting guide. This chapter offers an overview of following chapters and defines the approach taken. It also gives a variety of definitions of gaps and a quick method of reference to these gaps. This chapter further introduces the authors and their backgrounds.

Key Words

Gaps; Information Technology; Software

1. Introduction

The effectiveness of manufacturing depends on information technology. This in turn depends not only on the hardware and software technologies, but the ability to apply these technologies in a useful manner. Both hardware and software are advancing at a tremendous pace. However, the effective application of these innovations is lacking in manufacturing enterprises. This problem can be solved in large part by the utilization of software in creative ways. The lack of available software is the topic of this monograph. In this document we identify these lacks in terms of gaps. We do so with the purpose of presenting opportunities for software developers and manufacturing enterprises. These opportunities, when further clarified and acted upon, will provide benefits to the manufacturing industry in general, as well as specific rewards for the software developers and manufacturers who become directly involved.

1.1. Background. Since the introduction of Information Technology, more commonly referred to as computer technology, manufacturing enterprises have been one of the primary commercial beneficiaries. As early as the 1950s, software applications for the Department of Defense were developed to improve or extend manufacturing enterprise capabilities. As technology improved and costs have lowered, the acquisition of Information Technology has accelerated to an almost unbelievable rate. Within the United States today one would be hard pressed to find a manufacturing concern which does not utilize a computer in the performance of its work. Computer technology is on its way to becoming as prevalent in industry as the telephone has become.

Unlike the common telephone however, the computer is a more adaptive piece of technology. The component which makes this technology so flexible is software. Software is the instructions by which the computer performs the tasks given. It is through these electronically interpretable instructions that the computer receives its capabilities and limitations. This is the topic which sets the context of this monograph. This text is a survey of the current capabilities and limitations of software developed to support a manufacturing enterprise. Specifically this is a document about the existing gaps in software performance within these current capabilities.

The monograph is not meant as a historical or chronological document about the development of Information Technology or its deployment within the Manufacturing Industry. However, in order for the reader to develop an understanding for such a complex topic, some historical and technological tutorial material is enclosed.

The tutorial material enclosed within each chapter is meant to impart to the reader a conceptual level of understanding about the various functions, perspective and abstractions levels for software spans within present day manufacturing enterprises. Specifically, it draws the attention of the reader to those areas of unfulfilled requirements which represent gaps in capabilities. It is these gaps which represent opportunity for entrepreneurs within the manufacturing software industry.

1.2. Concepts. Prior to the main text of this monograph several terms need to be defined and mechanisms for understanding need to be introduced. These include the specific connotation of

the term gap, the framework used for the manufacturing enterprise and the various levels of abstraction and the perspectives utilized in the discussions.

1.2.1. Definition of Gap. Prior to a discussion about the various gaps in existence within manufacturing software, we should define what is meant by our use of the term and provide a taxonomy of the various types of gaps under study.

A gap according to Webster's Dictionary is a) a break in continuity b) a separation in space. While informative, this definition is not very specific as there exist many types of gaps: Functional, Connectivity or Integration, Scale, Operational Paradigm, and Platform. (See Table for Definitions)

Term	Definition
Functional GAP	A missing process, function or application capability.
Integration GAP	An existing application which lacks the ability to send or receive information to other applications in a processable manner.
Scale GAP	An existing application function or capability which is not scaled to the spectrum of business sizes.
Operational Paradigm GAP	An existing application function or capability which is not adaptable within different enterprise operating models.
Platform GAP	An existing application function or capability which is not present on major platforms.

1.2.2. Manufacturing Enterprise Framework. This text has segmented manufacturing concerns into a hierarchical series of abstraction levels. Within each successive chapter, the reader has been taken to lower levels of abstraction. At each higher level of abstraction, a broader but less detailed amount of conceptualization/focus is applied in dealing with manufacturing. This Hierarchical Framework parallels the SME and various other conceptual models of manufacturing concerns.

At each level of conceptualization of a manufacturing enterprise different issues are of concern. These are the issues defined by each level of the enterprise that are opportunities for software developers in identifying solutions. To those at the management level, issues of an enterprise's operations extend to include vertical as well as horizontal issues. However until recently, examination at the detail level, has not been done, either by the supervisory personnel or upper management.

It is between as opposed to within these layers, that the software industry has not focussed during the past years. As such, it is this area of visualization and integration tools where significant gaps exist. Such functions as data integration enablers, workflow control, configuration management and data management are not yet fully functional and no such software products have been fielded.

In general software within the manufacturing enterprise functional architecture exists with various capabilities; however, the porting of the applications does not exist on the needed platforms. Scaling or sizing is not to the appropriate size business, or connections via data or functions are missing. Gaps translate to market spot gaps versus functional gaps.

1.2.3. Abstraction Level. Within a typical enterprise exists a hierarchy of management, control and execution levels. An abstraction level is a scoping mechanism which limits the domain under examination and sets the context. At each successively lower level, the granularity of concerns becomes more detailed and less global in application upon the enterprise as a whole. An example of this, would be the differences in decisions a CEO would make versus those of a shop floor department foreman.

1.2.4. Information Technology Perspective. An Information Technology perspective views an enterprise not from the business objectives or tasks point of view, but rather through the application of various information manipulation tasks. These include: Data Management, Communications, Configuration Management, Semantic Integration. While many of these functions are similar to business tasks of the same names, the objects under manipulation are different. In the case of the manufacturing enterprises, the objects under manipulation are the physical products to be produced. In the case of Information Technology, the objects of focus are representations of information which may be the same physical products that a manufacturing enterprise operates upon.

1.3. Monograph Team. This monograph has been produced under the direction of SMS Software Management Incorporated, which contributed and coordinated the work of several experts in the fields of software and manufacturing. Their experiences range from small software development houses to large corporate manufacturing concerns. As such, the contributions of each provide a unique diversity to the material presented. This document is not a single perspective, but an integration of several perspectives from these experts in the field. In addition opinions were solicited from other leaders in both Software and Manufacturing Industries.

The effort has involved the thoughts of a total of nine experts throughout the United States. Seven of these experts were actual authors and two were independent reviewers. Their contributions were:

Authors:

Dennis Donohue
SMS Software Management Incorporated

Brian K. Seitz
B.K. Seitz & Associates

Gil Laware
Management Support Systems

Rick McHard
SMS Software Management Incorporated

William Burkett
Product Data Integration Technologies

Mark Halpern
D.H. Brown Associates, Inc.

Seth Hunter
Miles Diagnostic Laboratories

Reviewers

Bill Bergiadis
McDonnell Douglas

Bruce Navsky
SMS Software Management Incorporated

The document was produced over ten months, from different locations across the country, using different word processing equipment in separate but coordinated efforts. This material was then electronically collected and integrated into the text which you are reading. The team utilized distributed capabilities available to support electronic commerce now.

2. Monograph Organization

This monograph is organized to present the current technology and gaps in four different areas or domains, which comprise chapters 2 through 5. Information and the identified gaps are evaluated and presented in Chapter 6 to show the opportunities existing for software entrepreneurs.

2.1. General Organization. The four main chapters (2 through 5) discuss specific as follows:

Chapter 2

Discusses the requirements that software developers must understand - how an enterprise, process, or product is constructed. This is done through the introduction of the concept of models. These conceptual tools allow an individual to analyze and design objects.

Chapter 3

Discusses the functional and operational roles software plays within a manufacturing

enterprise. The author explores several alternative architectures and directions for manufacturing systems to take in the future. This chapter develops the contexts for the following two chapters.

Chapter 4

Presents the concept of integration before the reader and the gaps within currently fielded products as well as tools to aid in their mitigation. The author suggests a method by which small and large enterprises may evaluate and identify the integration gaps to be addressed.

Chapter 5

Introduces the potential developer to functional and integration gaps within the engineering design and data management domain. Within this section, a reader will learn of a series existing gaps that are typical in current manufacturing enterprises. The author details where functional capabilities are lacking that need to be programmed, and where new technology needs to be developed.

2.2. Guide to Find Existing Gaps Described in the Text. What types and where Gaps can be found throughout each of the major chapters, refer to document and chapter organization section within this chapter. (Table follows).

Chapter	Function	Integration	Platform	Scalability	Paradigm
Chapter 2	4	4	4		
Chapter 3	4	4	4		4
Chapter 4	4	4		4	4
Chapter 5	4		4	4	

2.3 Chapter Organization

The major chapters within this monograph follow the following model of organization:

- Introduction
- State of the Art description
- Gaps within the Art
- Conclusions

The introduction provides the reader a general discussion of the topic. The topic introduction familiarizes the reader to the perspective this chapter author brings to the subject of software gaps. From this section introduction, a reader should get a general feel for the area under examination and the perspective from which it will be further discussed. This subsection provides the context from which the chapter is built upon.

The next chapter section presents to the reader either a state-of-the-art or state-of-the-Industry discussion. This is not meant as an all inclusive examination of the topic, but an overview. The results of such an overview is a further refinement of the context introduced in the earlier section. The prior materials are designed to provide the reader with the tools needed to relate the various elements contained within the chapter together.

The third section focuses upon the identification, definition, and categorization of gaps according to the perspective introduced earlier. This provides a survey of the gaps identified using this particular view of software. It is this one subsection in each main chapter that carries the primary objective of the monograph. Until now all the materials presented establish a foundation for the reader to understand the information contained in this section.

A conclusions section is at the end of each of the major chapters. Within this section the author summarizes the material presented and draws conclusions.

3. Conclusion

During the last two decades software for product design has taken a tremendous advance. Computer software to aid the engineering function has switched from merely an aid in calculations, to an accurate drafting pencil, to an environment for modeling a product in a simulated reality. With each technological advance the requirement for management controls has increased. Today this is an area of concern within the engineering function. A significant series of gaps exist in the control of the engineering process and the management of its' data.

This monograph should not be taken as an exhaustive treatment, but a survey of the areas of gaps in the market for industrial software. It presents material to familiarize the potential developer about the broad area of opportunity within the manufacturing software market. The document has defined early what is meant by a software gap. Further, a taxonomy of specific gap-types and attributes is introduced so that the reader may evaluate and qualify various potential opportunities. Where possible the differences between functional versus market/spot gaps (which are called platform gaps) have been addressed within this chapter.

The monograph covers software gaps at several different levels and vantage points:

- Enterprise Visibility Gaps, i.e., Modeling and Simulation Software
- Operational Gaps, i.e., Application Software or Enterprise Functional Architecture
- Inter and IntraEnterprise Communication Gaps i.e., Translation and Communication Software (Enabling Manufacturing Enterprise Integration)
- Manufacturing Enterprise Design and Data Management

This yields a unique perspective to the reader not usually found in survey material. Overall the document attempts to present the broad picture of the field of gaps in manufacturing software.

However, the seamless capabilities to perform such, are not yet realized within the current suite of software supporting manufacturing today. Special attention to this one point should be given as one reads through chapter 3 Enabling Manufacturing Enterprise Integration.

About the Authors

Dennis J. Donohue is the founder and president of SMS Software Management Incorporated. He has twenty four years experience in the use of state-of-the-art software development technology in the applications of real-time process control and distributed management information systems. Under his direction, SMS has provided development of custom software products for a world wide client base of end users and original equipment manufacturers.

Brian K. Seitz is an independent consultant and senior member of CASE/SME with over fifteen years of experience in information technology for manufacturing. He has been active in the field of manufacturing software standards and development for firms such as IBM, Rockwell International, Lockheed, and McDonnell-Douglas. A guest lecturer at AUTOFACT, IBM Guide/Share and AIAA Design and Operations Conferences in the fields of Enterprise Architecture and Integration, Electronic Commerce, and the application of information systems standards and technology.

Chapter 2

Gaps in Modeling for Manufacturing Enterprises

Gilbert W. Laware
Management Support Systems, Inc.
55 Summersweet Place
Stratford, Connecticut 06497-1078
203/380-2344

Abstract

Today's challenges have driven industrial enterprises to closely examine the operations of delivering their products and services. Enterprises are examining their operations by characterizing why and how work is done and what changes can be made to improve overall performance. In the examination, key operational characteristics include evaluating their manufacturing cost position, managing the technological sophistication of the production (labor, equipment, and facilities) elements, designing for flexibility to adjust the products and services in very short lead times, and obtaining low cost raw materials. A critical element in executing these operational activities is understanding how information technologies can maximize the strengths and reduce the weaknesses of the enterprise. New technologies affect business strategies, organizational design, decision-processes, the actual information and work flows, and ultimately, how people work.

There are significant gaps in the information technology available to meet these challenges. Each gap affects the others in providing solutions for the company. Specifically, gaps exist in the tools currently used to portray business relationships and information about those relationships. There are also no means to easily assimilate different modeling methodologies, to align business and technology planning systems and, finally, to absorb new technologies into the manufacturing enterprise.

Key Words

Architecture; Business Model; Data Model; Function Model; Information Model; Knowledge-Base; Model; Model-Based Systems; Process Model; Reengineering; Reverse Engineering

1. Introduction

An Overview: Manufacturing businesses need a set of robust models to respond to problems with the right technological solutions. Manufacturing is becoming increasingly dependent upon model-based systems. Different models (action--process, culture, data, document, decision, event, financial, information, location, organization, person, project, risk, system, etc.) define the knowledge-bases of the enterprise. Modeling is one means to help integrate these knowledge-bases and to ensure the development of highly integrated data, communications, methods and tools to meet the contemporary problems faced by manufacturing enterprises in the 1990s.

These problems are global, since industrial enterprises are making worldwide alliances to produce products and services faster, with better quality, and at reduced costs. The use of information technologies is one driver of change.

Industrial enterprises are closely examining the operations of delivering their products and services. Demands for high quality, long-term reliability, low costs, quick changeovers, small manufacturing lots and short-lead times have challenged each manufacturer to examine some key operational characteristics. They are evaluating their manufacturing cost position, managing the technological sophistication of the elements of production (labor, equipment, and facilities), designing for flexibility to adjust the products and services in very short lead times, and obtaining low cost raw materials to meet these demands. Enterprises are examining their operations. They are characterizing why and how work is done to improve the performance levels of their business. A critical element in executing these operational activities is understanding how the business uses and can use information technologies to maximize the strengths and reduce the weaknesses of the enterprise.

New information technologies affect business strategies, organizational design, decision-processes, the actual information and work flows, and ultimately, how people work.

From an enterprise perspective, gaps exist in several areas. The gaps are:

1. There is no easy means to portray the business relationships and information about those relationships.
2. There is no easy means to easily assimilate various modeling methodologies with these new technologies.
3. The alignment of business and technology planning systems is challenging.
4. The absorption of new technologies into the manufacturing enterprise is difficult.

In manufacturing enterprises, information technologies have helped in many ways. They have successfully improved the flow of raw materials from supplier to manufacturing operations and the subsequent delivery of an end-product to the enterprise's customers. In the development of new product designs, Computer Assisted Design (CAD) reduces the time required to deliver resulting products to market. Computer-driven numerical control

operations reduce wasted raw materials while increasing operational efficiencies but, they are used in functional areas within the enterprise. They do not fully address the gaps in the knowledge needed for enterprise management, which is why the challenges of the 1990s are different.

From a software technology perspective, the legacy of current manufacturing software applications is functional and vertical in structure. Legacy applications often inhibit rather than enable new manufacturing capabilities. They provide the advantages of being focused and very effective for a specific function. The disadvantage is that they are narrowly defined. They lack sufficient semantics about the business, its data, rules, and organizational usage to be easily linked to other enterprise systems and technologies. Therefore, contemporary methods require a different process to plan, structure and deliver information technologies. The challenges of the 1990s will drive this change since flexible information technology support is a requirement for both large and small manufacturing enterprises. Flexible information technology becomes the basic requirement for discussions of model-based systems.

20th Century's Organization Setting: We need to assess today's organizational environment. The virtual organization deals with an environment which is very different from the one we have experienced. Although the functional nature of an enterprise will remain, its implementation will be vastly different. There are three major reasons for this change:

1. New technology will enable world-wide communications to exist within seconds (using new communications and data compression techniques).
2. The speed and capabilities of computers will increase the speed of business transactions (using parallel processors with new storage and computing techniques).
3. Computing systems will be easier to use and have human-like communication capabilities (using different combinations of animation, graphics, sound, and text).

Since each business can outsource many of its operations to an independent vendor or supplier, the communication between service providers will be driven by offers and requests through computerized communication systems. Thus, the speed of providing the product and service will depend on the skills, capabilities and technological strengths of the corporation and its many partners.

The competitive environment: Today's manufacturing business:

1. Faces increasing competition in a global economy.
2. Has to produce valued product and services.
3. Needs to use technologies to enhance their competitiveness.

These drive new business requirements.

Business requirements are statements of software requirements: Manufacturing executives are establishing clear statements outlining what they believe the future state of their

business will be. These statements help to set up the expectations and the respective relationships that make the business run smoothly. Goals define the period in which the business intends to accomplish specific actions. These become part of the overall business model.

Enpowered people need to anticipate and formulate the right product and services at the right time for the right reason that requires this knowledge about business. Table 2.1 shows some business things (objects) and the possible relationships that can exist when these questions are answered. This list is not inclusive but it illustrates many key objects that are modeled in a manufacturing business. Since additional groupings exist under each listed object, this serves as a useful basis for thinking about knowledge-based business integration.

Each of these business things (objects) is important in the planning, organizing, directing and controlling the business. The deliverables, products and services are the focuses of the business' operational activities. Other relationships exist that support the production of the products and services the business markets and sells. Work flows exist using these objects and are the infrastructures of how the business works. These relationships link two or more objects in a way that structures the culture of the business.

Things \ ?'s	WHAT	HOW	WHO	WHERE	WHEN	WHY
Strategies						
Missions / Goals						
Organizations						
Assets ($,Machines,etc.)						
Locations						
Functions / Processes / Activities / Tasks / Events						
Products / Services / Deliverables						
Decisions						
Rules						
Information (Entities)						
Flows						
Systems						
Projects						
Datum (Element of Data)						

Table 2.1

Different approaches are used to help represent the relationships for business decision-making. They require differents levels of granularity depending upon the reason for the decision. In aggregate, these representations are the intellectual property of the business. Using this as a basic framework, we will discuss how modeling portrays this knowledge-based integration and the gaps that exist in producing software which reflects the contextual knowledge of the manufacturing business.

Uses of Information Technology: Advances in technology are continually making it easier for organizations to realize improvements. Michael E. Porter states:

> Technology change is one of the principal drivers of competition. It plays a major role in industry change as well as in creating new industries. It is also a great equalizer eroding the competitive advantage of even well- entrenched firms and propelling others to the forefront. Many of today's great firms grew out of technology change that they were able to exploit. Of all the things that can change the rules of competition, technological change is among the most prominent.[1]

Scanning various science-based technological developments (i.e., electronic, mechanical, computer-based) that are potentially applicable to a business' product or service is a business necessity. This is an outward focus. When a business scans these technologies, it may also want to apply some of them to its own organizational environment. This is the case with the technologies associated with information systems.

Different technology changes in storage capacities, computing speeds and telecommunications capabilities have provided new functions in hardware, software and communications. This has expanded the capabilities to do: Client/Server, Groupware, Imaging, Multimedia, Parallel Processing, Wireless Communication and Virtual Reality. These represent only a sampling of the changes in the information technologies. As the fundamental capabilities in storage capacities, computing speeds and telecommunications change, additional functions will affect the possibilities for manufacturing systems integration.

Need to transform the Manufacturing business: The approaches for changing a manufacturing business were clearly summarized by the study on The Competitive Edge: Research Priorities for U.S. Manufacturing[2]. It identified five critical areas. First toimprove equipment reliability, decrease cycle times, and achieve the greater precision dictated by demands for higher quality, manufacturing must adopt and further develop intelligent manufacturing control. Second, manufacturing must maximize the productivity of its capital investments through improved equipment reliability and maintenance practices. Third, manufacturing must enhance product characteristics, for instance, by using advanced engineered materials to reduce weight, broaden service temperature capabilities, impart multifunctionality, or improve life-cycle performance. Fourth, to speed time to market, manufacturing must employ product realization techniques and adopt organizational changes that foster effective use of these techniques. Finally, creation of the new work force --highly

---

[1]M. E. Porter, Competitive Advantage Creating and Sustaining Superior Performance (New York: The Free Press, 1985),p.164.

[2] ____________, *The Competitive Edge: Research Priorities for U. S. Manufacturing*, National Academy Press, Washington, DC(1991), p. 9.

adaptable and possessing multidisciplinary skills of high order--that is critical to the application of these techniques and technologies will require a focus on manufacturing skills' improvement. These approaches drive the integration of enterprise operations.

Integration of information systems is a critical part of changing the systems supporting some manufacturing operations. Advanced manufacturing technology will seldom yield the anticipated flexibility and/or productivity in an organization unless corresponding changes are made in the organization itself and in its information systems and resources[3]. Change requires a careful recognition of the organization's culture and the objectives sought by the change. Building a commitment to change requires understanding the steps, involving the people who need to commit to change, and investing time and resources. One way or another, it is an emotional and a monetary expense. One underlying assumption across these five areas of manufacturing is that a highly cooperative linkage between people, information systems and machines exists. Thus, an organization wide knowledge-base requires data and data structures, decision and decision structures, people, process structures, organizations and their structures, and various time dimensions covering many planning horizons. This knowledge-base uses many information technologies mentioned previously and is a significant change to the business' culture and organization.

2. Modeling State-of-Art

Models specify, map and profile the business by making explicit the definitions, assumptions, relationships and rules associated with a particular subject area. Thus, models become one means to communicate how the enterprise plans, organizes, directs and controls its environment. They provide a means to understand and relate to some part of the reality of the environment in which they are operating (Figure 2.1).

Model Definition: A model is a representation of something. It isn't the actual thing but is similar since most of the key characteristics of the thing are being represented. For example, we give children toy dolls and trains to play with; they are physical models.

In different disciplines, the definition[4] of a model has different meaning. It can be:

(1) A standard or example for imitation or comparison.
(2) A representation, generally in miniature, to show the construction or serve as a copy of something.
(3) A pattern or mode of structure of formation.
(4) A system of things and relations satisfying a set of axioms.

Therefore, a model serves a purpose. The purpose dictates the level of generalization or detail representation necessary to express the desired purpose. We want the model to serve as an analogue to the thing being examined.

---

[3] Ibid., p. 10.

[4] J. Stein, Editor, *Random House Dictionary of the English Language: The Unabridged Edition*, Random House, Inc., New York, NY(1967), p. 920.

Models -- Specify, Map and Profile

Specify by:

Making explicit statements about:

Definitions of Concepts and Assumptions

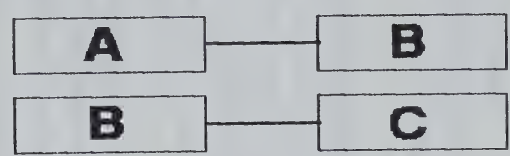
Specific Relationships

Rules associated with Relationships

Map by:

Defining specific relationships

Linking two things



Showing linkages across many relationships (Inference)

Profile by:

Providing insight about:

A particular area being examined

The differences due to changes in states

The effects of different decision criteria

There are many forms of models: descriptive, mathematical, pictorial (graphic), or physical. Several types of models are shown in Figure 2.2.

A narrative model describes some object, activity or phenomenon in spoken or written form is a Descriptive Model. A mathematical or quantitative model uses mathematics to describe the relationship of two or more things. Any formula is a mathematical model. familiar with quantitative models. Mathematical models are the basis of various applications in stress analysis, materials requirements planning (MRP), distribution, manufacturing resources planning (MRP II), engineering design, logistical, plant and shop floor control systems. Many models assess what something should look like in the future are known as normative or prescriptive models. A pictorial or graphic model is a two or three-dimensional representation of something. In manufacturing, charts (bar, line, pie), diagrams (sequence, flow, layout) and images (maps, transportation, and machines) illustrate such these models. A financial model represents something in terms of money. A physical model is a tangible three-dimensional representation that exists. In the airplane and automotive industries, computer-aided design (CAD) models develop specifications for the new product. Scaled-down versions of product are developed to examine different physical characteristics of the product. A statistical model explains the relationships and causation between the observed data.

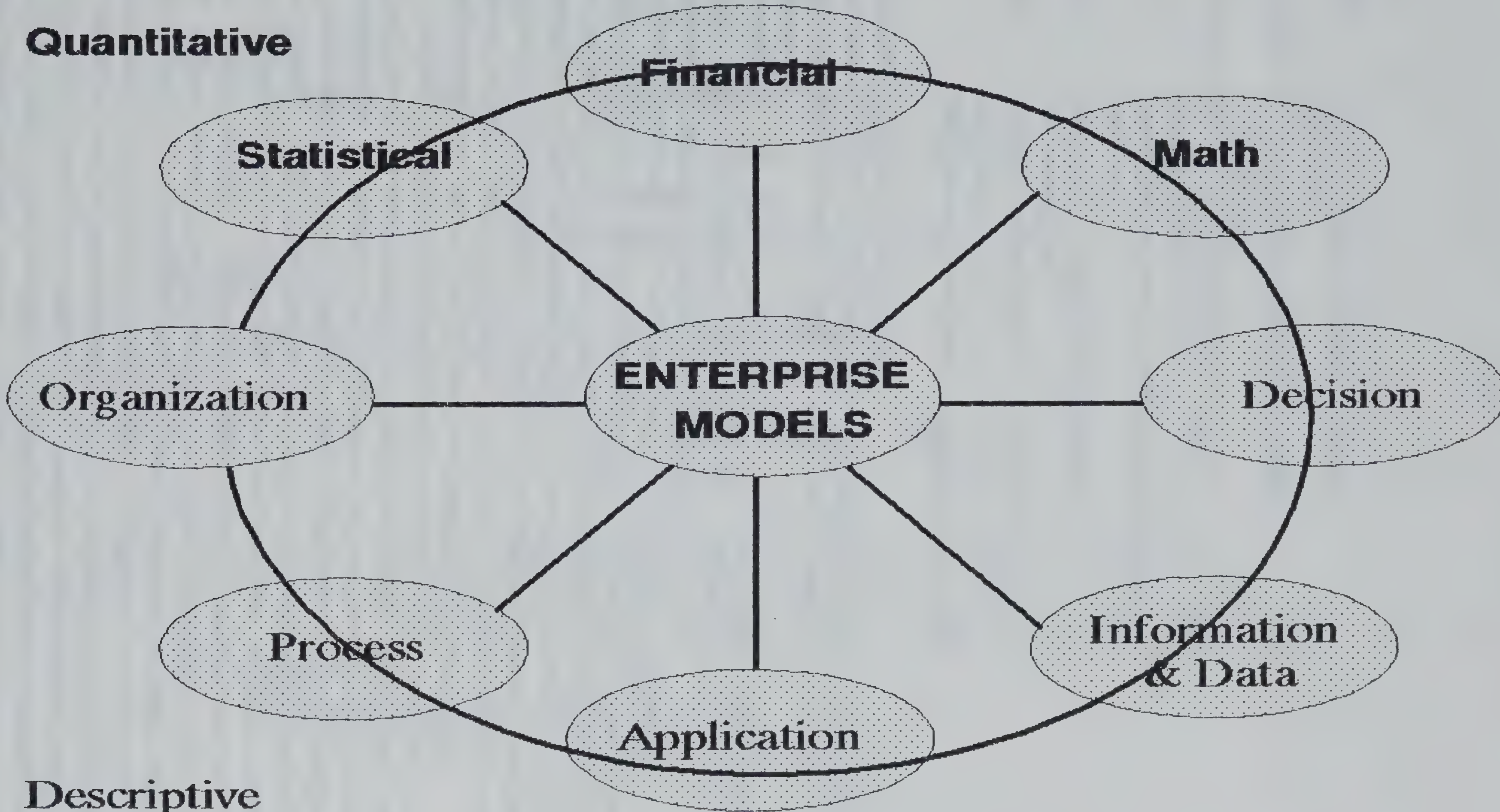
Each model is not necessarily independent of each other. Usually, a descriptive model precedes the formation of another model. In this sense, individuals transform the description into another quantitative or pictorial representation. Various models are combined to show alternate views, for example, a CAD Model contains both a mathematical and pictorial model. Network analysis techniques help planning and controlling of projects by illustrating the flow and sequence of project activities. It combines several means to show relationships (a graphical representation and their numerical dependencies).

Models are tools that aid decision-making. As such, they are also classified in terms of time (static or changing), risk (deterministic or probabilistic) and predictive (optimized or sub-optimized) capabilities.

Considering the dependency of relationships that exist in models, it is no different for the information systems arena. The Figure following Descriptive and Graphic Models are used in the information systems. They include:

Data -	Describes the information, relationships, and data being used.
Decision -	Express critical conditions or rules about a set of choices (actions).
Function -	Describe how activities or actions are executed.
Event -	Describes the time and things that cause some change of state.
Information -	Describes the information and key relationships.
Organization -	A static representation of the business structure.
Process -	Describe different actions on objects.

Types of Models



For example, there are different views or categories of Process Models. They include a Business Process, a Task Plan, a Business Function and a Functional model. One type of model is a Business Process Model. It examines a logically related set of activities, jobs, tasks (actions and objects) to achieve an outcome, result, or deliverable (e.g., build an airplane). A contemporary purpose is to reengineer (create a new or redesign) an existing process. In this sense, processes are cross functional and cut across organizational boundaries. A process is a specific ordering of work activities across time and place, with a beginning, an end, and clearly identifies inputs and outputs: a structure for action[5].

In the development of the information system's arena, The designer divides the activities the application system is to do into understandable parts. Each part does some "what" or functions (e.g., Produce Part Design). Functional decomposition or stepwise refinement is the work activity a designer does. He/she specifies the order of the interrelated parts (modules) or actions will be done to accomplish a desired result or outcome (how). Each part is logically related. Each part clearly defines the interconnections between other parts and the total system is divided into understandable parts by showing greater levels of detail. Figure 2.3 shows an example of a function model representing a manufacturing business graphically as a hierarchy.

It shows the major actions this business believes critical to its operational success. As a Descriptive Model, it reflects a consensus about the nature of this manufacturing business. Depending upon the culture and experience of the organization, one functional model within the same industry is not uniformly applicable. The management and actual teams who develop these models describing their actual operations affect the points of emphasis the business places a focus. The model emphasizes those major functions and the sub-functions. Several models have been defined to more than 30 different levels of detail.

The next Figure 2.4 shows a different form of a functional model(IDEF0). This figure shows three different levels. The first level (Level 0) shows the functions performed by this business unit. On the left side, a description of the functional context is shown. Correspondingly, at this level, there are specific objectives the unit is trying to accomplish. They are shown on the right. Within the diagram, the inputs, outputs, controls and mechanisms are shown. With each suceeding level, the amount of detail increases. But, it contains more detail about an item shown at a higher level. The two functional models shown in these diagrams provide you with two different perspectives. The first shows a broader perspective of the whole business; the second is more specific.

Answering the question "why" is a key element in the development of any model. An enterprise models can consist of one or more combinations of the types of models (Figure 2.5). Therefore, a combination of perspectives are necessary to represent the business operation. From an operational perspective, businesses' are using aggregate models to describe their business from differing perspectives. Different models effect both logical and physical changes to the business' operation. From an information technology perspective,

---

[5] T. H. Davenport, *Process Innovation: Reengineering Work through Information Technology*, Harvard Business School Press, Boston, MA(1993), p. 5.

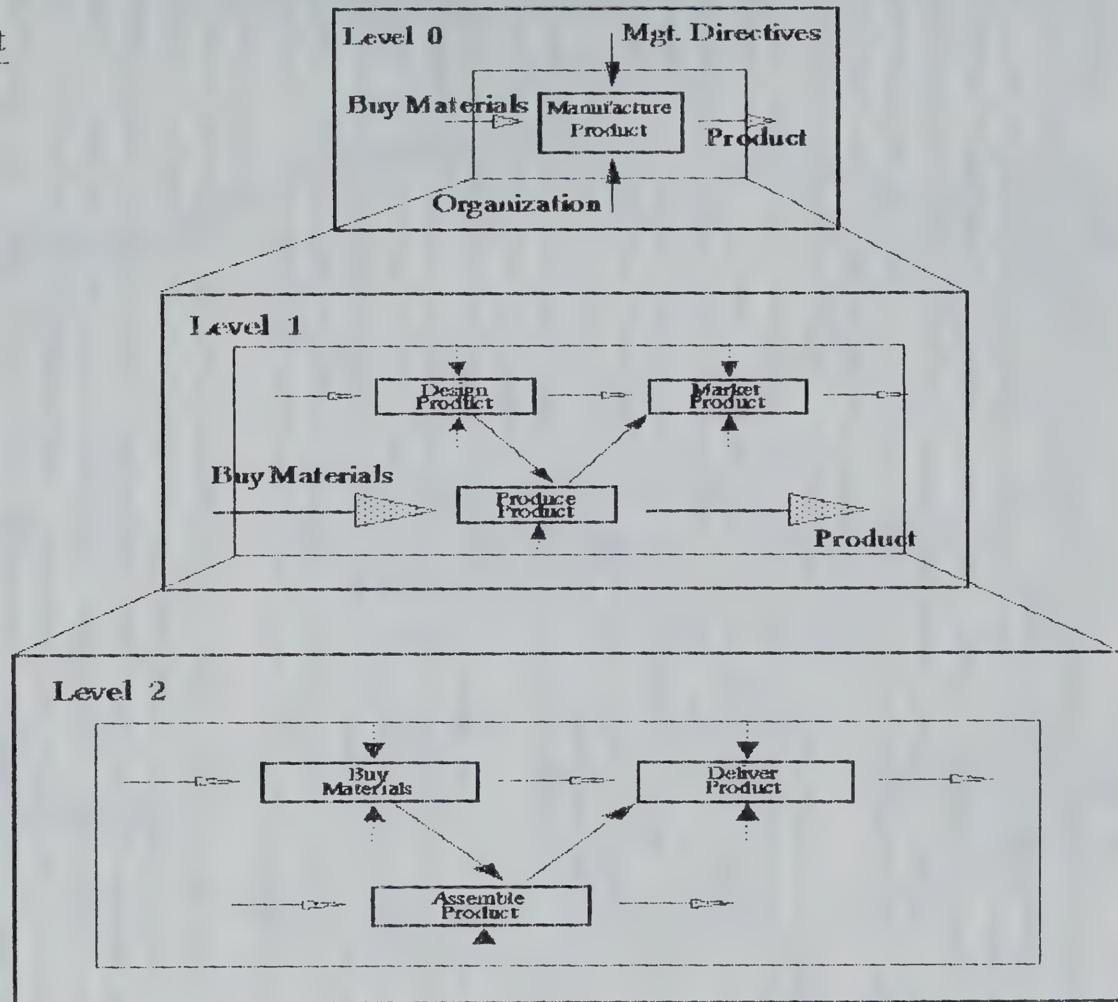
Functional Decomposition

Leveling Context

The Business

Manufacturing

Assembly Operations



Business Objecti

Increase Market S
Reduce Cost

Reduce Labor Co
Lower Material C
Use JIT

Reduce Scrap
Increase Quality
Use Robotics

logical models affect the business, information, data model, and specific application software models. As a result, the existing technology model is changed or a new technology (database, communications, etc.) is put in place to achieve the business objectives.

Considering the Business Model from a software requirement standpoint, it encompasses all the data, functions (actions), locations, events, rules, and rationale for the work flow change. Again, the scope and boundary of the project dictate the relationships that will be examined. The Information (Object or Entity) Model describes the key things used by this work.

As the project develops, a key-based Data Model shows the Figures 2.3 and 2.4 essential data work manages. Each data item is logically associated or attributed to the real-world objects shown in the Information Model.

The result is a specific application software model that will enable the new flow. If the flow affects many objects, there is an opportunity to link many models together to support the operational actions of the new flow (i.e., payroll, inventory, shop floor scheduling, process plan development, bill of material processing, etc.).

The Technology Model shows the information technologies used throughout the business and their relationship to the specific flow. It specifies the technologies that will be used for the desired application software system. It bounds the structures, languages, and rules needed to transfer the previous models into the specific set of technologies.

Within this application software development process, there are many key relationships. Each part contains a set relationships that need to be logically designed. There are two levels of analysis. One describes the logical relationships while the other deals with the physical use of those relationships. Each is a descriptive model (Figure 2.6). The Where points to location(s). It drives the client/server model for the distribution of function and data. The When highlights the time and conditions about the Figure 2.5 and 2.6 data which are required.

How describes a set of actions (Function, Method, Process, Procedure, Subroutine, Step) required to perform work with the data. What is the Objects used in the work flow process and the relationships they have to each other. The relationship between objects links the views of data needed to support the action between them. These are known as business rules. Underneath each object (Entity) and object relationship, the data characterizing the object is shown. These are the data structures and relationships that are included in the physical database models. The data definition language (DDL) defines the data structure and the data manipulation language (DML) supports the Function and behavioral characteristics of the object as described in the relationship.

The requirements process is the process of determining and knowing what the business wants. It is an unambiguous description of what is wanted -- often called a set of requirements or a problem statement[6]. It results in a set of intersecting models.

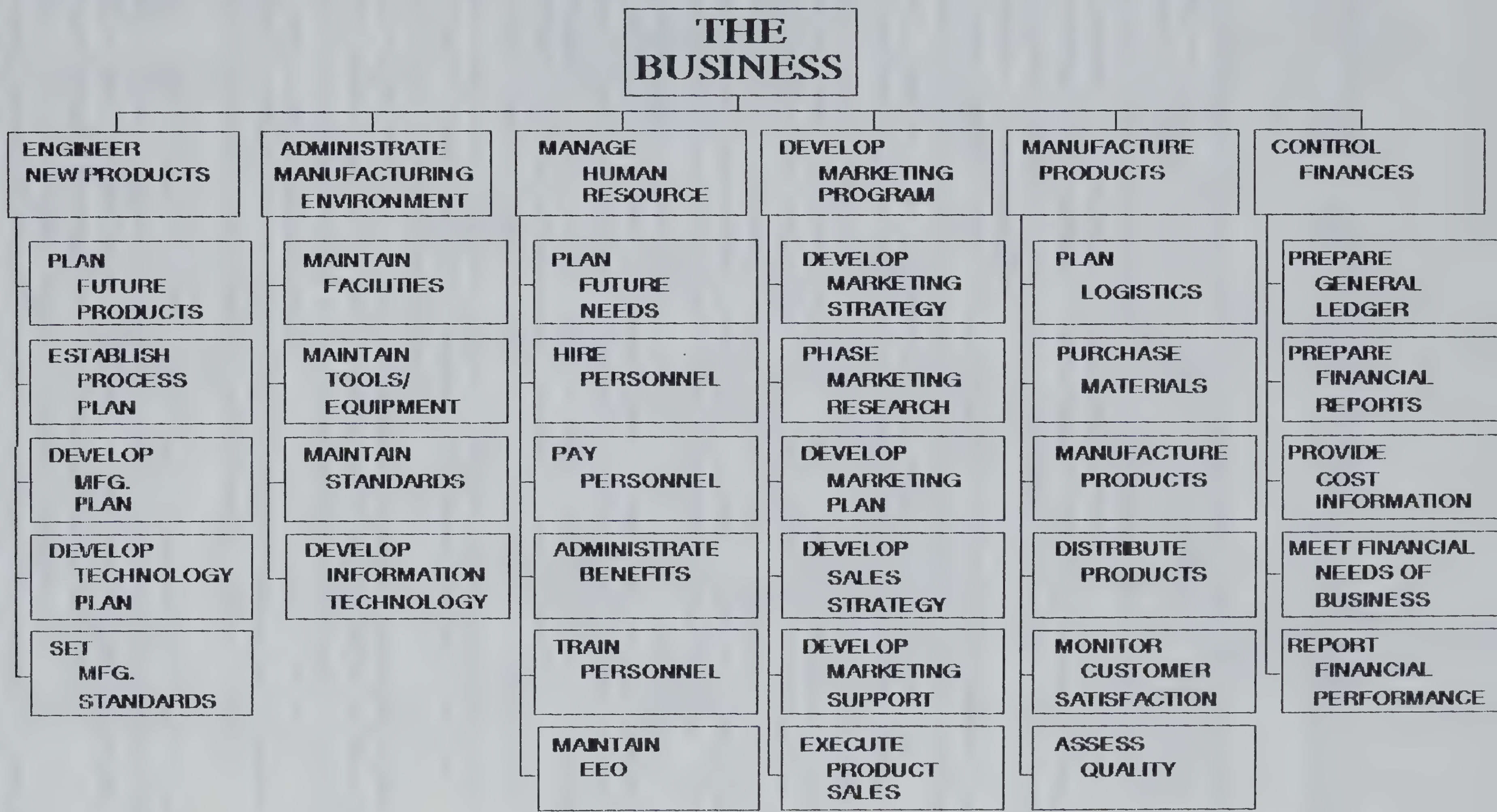
Descriptive requirements and models consist of other submodels dealing with the events, processes, information, data, rules, decisions, costs, flows, locations, organizations, risks, documents and risks (Figure 2.7). Each has a purpose and rationale. When used to produce part of an application software system, other descriptive requirements deal with the

---

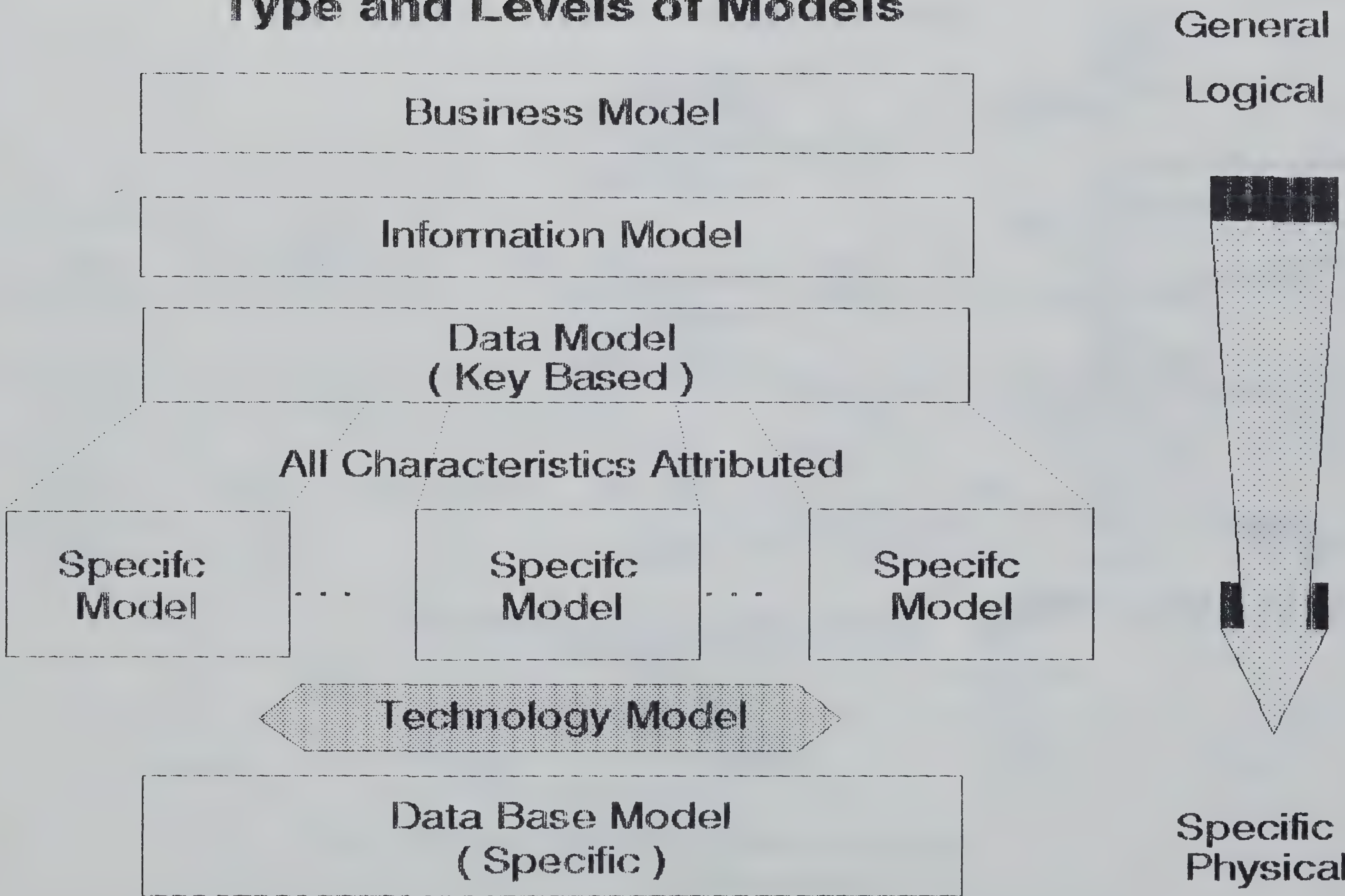
[6] D. C. Gause and G. M. Weinberg, *Exploring Requirements: Quality before Design*, Dorset House Publishing, New York, NY(1989), p. 1.

A FUNCTION Model of a Manufacturing Business

An Example



Type and Levels of Models



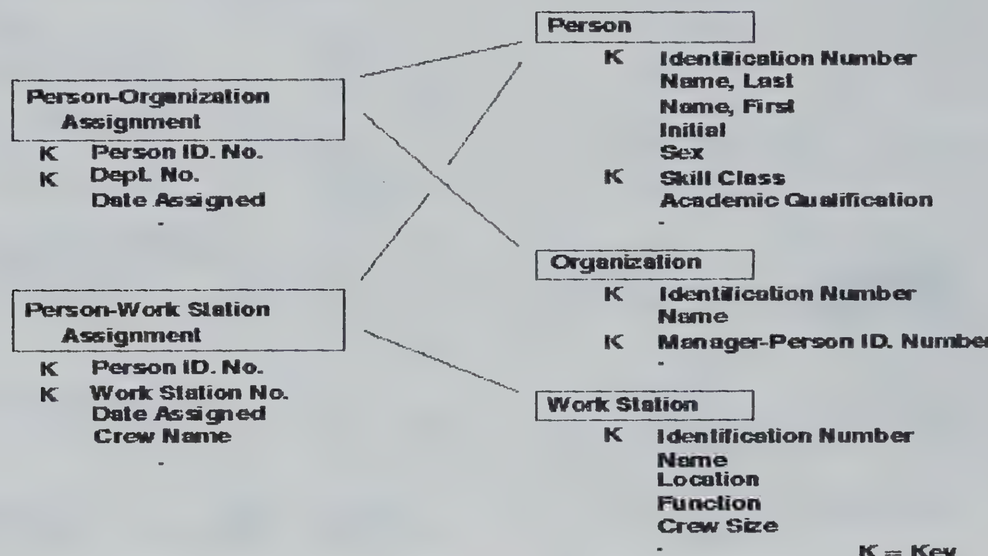
Descriptive Model Relationships

WHO
PEOPLE

WHY
MOTIVATION

Create Models

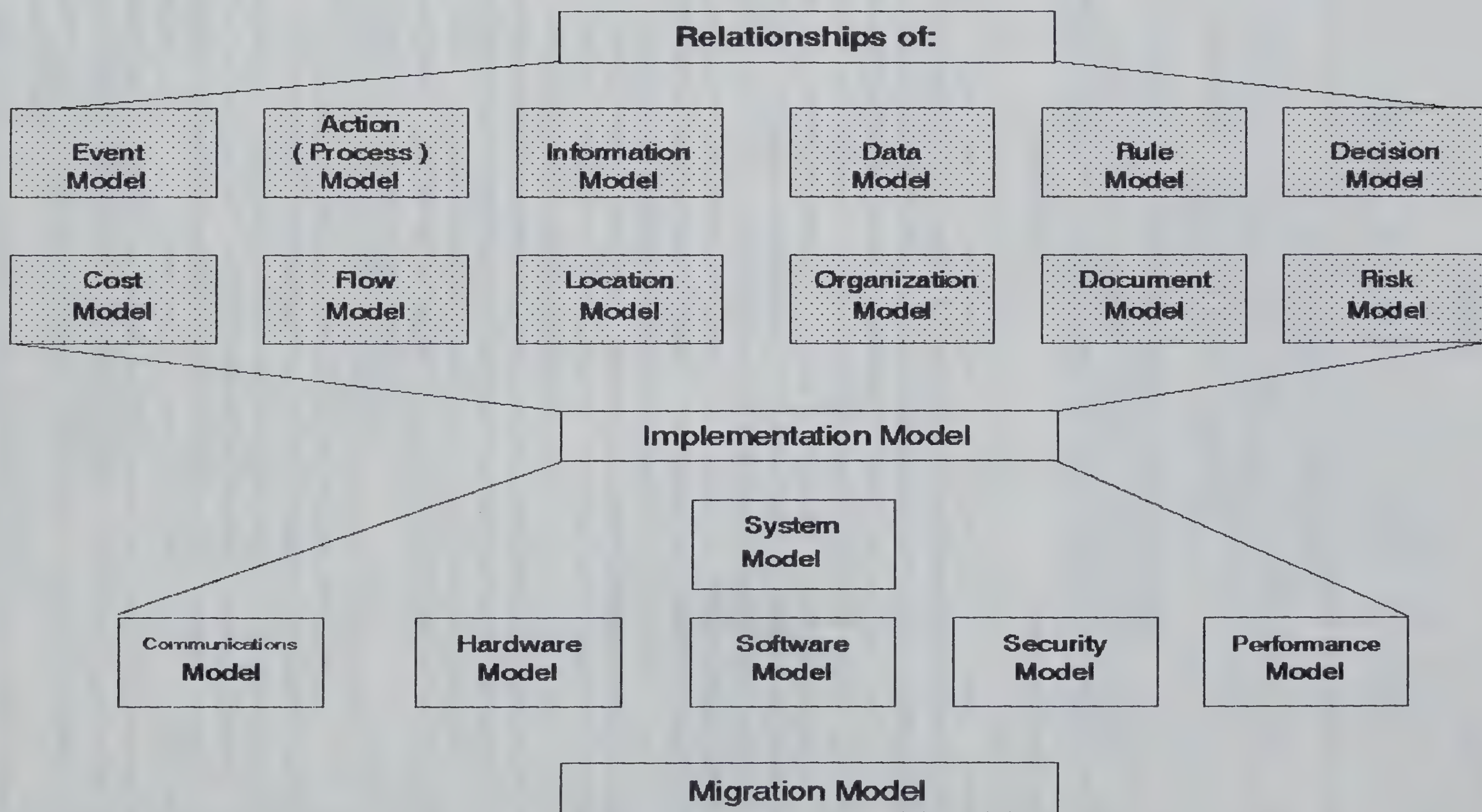
WHERE	WHEN	HOW	WHAT			WHAT
Location	Time	Process	Object	Relationships	Links	Object
Factory A	Weekly	Operate Work Cell	Work Station	Used by	1 or More	Person
			Person	Assigned	1	Organization



Show Business Rules

Different Models

deal with



communications, hardware, software, security and performance characteristics of the system. If the application exists today, the migration model defines the steps needed to move to the new application environment.

This gives you an insight into the number of relationships that exist in developing an application software system. The sheer volume of relationships is extensive! The number of relationships that must be examined is the reason old applications software migrations are so difficult to reverse engineer.

Understanding these business relationships is required since the business needs to: evaluate its strengths and weaknesses, link customer and business objectives for planning, combine different models to fit different solution sets, produce the right applications for the right purpose, and improve the application software systems productivity to dynamically fit with the changing business environment. Preparing to meet some of these needs, a manufacturing business may use several business approaches to plan, analyze, assess and deliver the right application software system solution.

Business Analysis Techniques for Modeling: A business analyst assesses a business situation in many ways. By doing comparisons, examining differences, listing options, determining casual relationships, and providing recommendations, the analyst supports the choice of the right solution in a particular business situation. In the manufacturing world, computer systems engineering is a problem-solving activity.

In business analysis, matrices are a common way to display one or more relationships. The matrices help classify, count, group, select, and rank data to understand the problem being analyzed. Here is a table showing commonly used techniques.

Many of these techniques are part of different methodologies. Computer-Aided Design (CAD) modeling provides a much faster way to think and design new products. This displays a principle of making a prototype design and refining it produces a better quality product. An engineer designing a part uses these techniques. In the examination of design, CAD data about the part's or similar parts characteristics are available in a database. The engineer orders, ranks, infers, clusters, and compares each part with other parts in the total design. The same principle applies to applications software development process.

The process of developing an application software system is similar to producing a new product or service for your business. It has a development life cycle and an operational and a maintenance cycle.

Technique	Descriptions
Clustering	Grouping an objects with others that have the same characteristics. Combine -- Create new model from selected model combinations Intersect -- Select and create relationships common to two or more models Select -- Choose a subset of one or more models based on specific object(s) or relationships

Comparing	Similarities -- Find out how models are alike. Differences -- Detect the degree in which one or more models are different. Full -- Exact equivalent of one model to other models. Partial -- Select from two or more sets of models.
Inferring	See Multiple Levels of Relationships -- Examining across two or more models based on particular object or relationship
Ordering	Grouping -- Ordering and selection of model Ordering -- Reversing ordering of model parts
Ranking	Rank -- Assigning a subjective value to an object or relationship in model.

The Waterfall model, shown in figure 2.8, shows the sequence of steps (actions) which are used to produce a software product. Each phase has a relationship with one or more models. Since the waterfall model is a sequential process, it results in:

Taking too long to produce the software product;
False assumptions about the requirements (statement of needs) as:
Not subject to change (stable and static) for the duration of the process,
Complete,
Traceable to the actual software instructions;
and a high number of errors due to translation process of the requirements into software instructions.

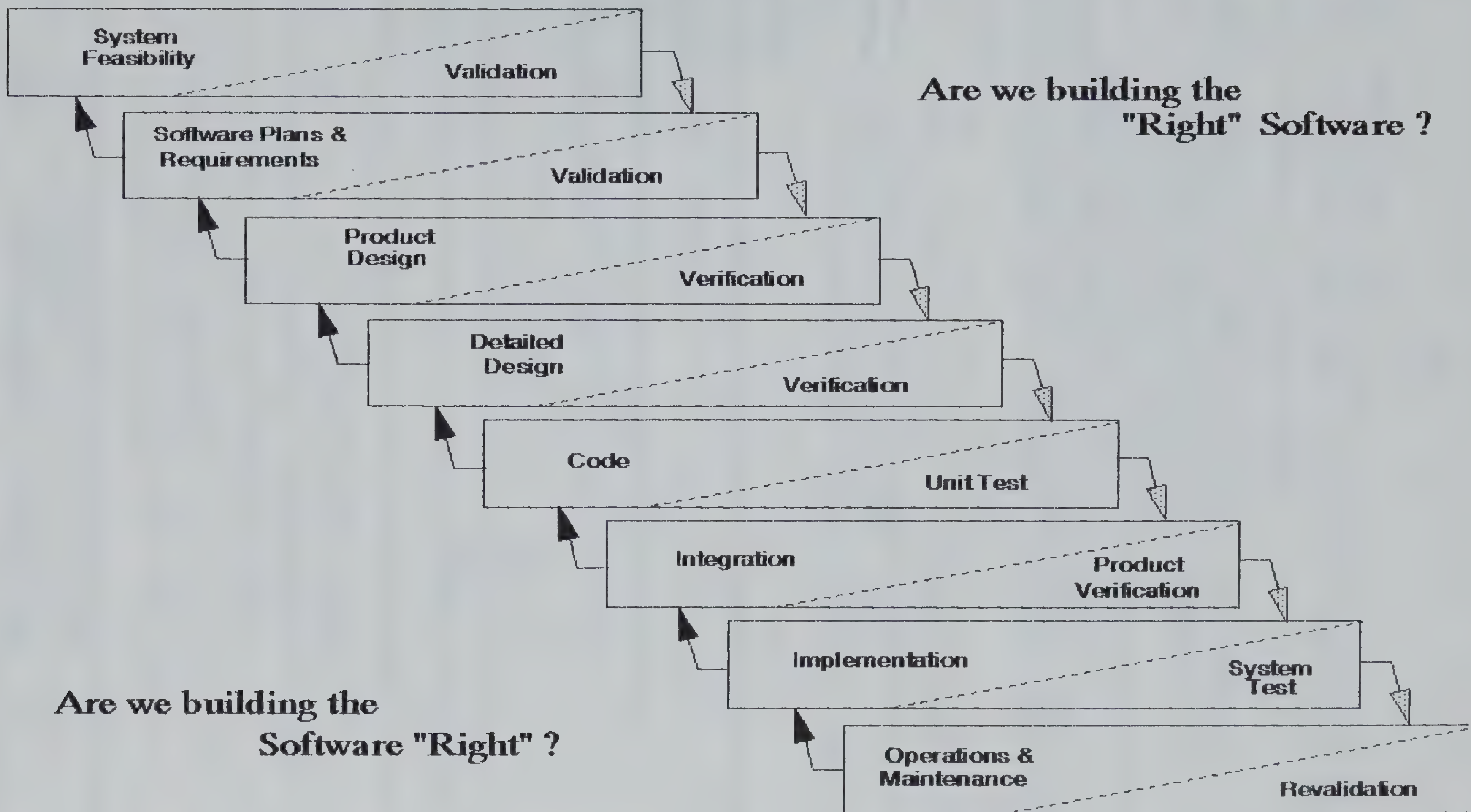
Alternative approaches to developing software are being used because of the inherent problems with the Waterfall development process for software[7]. They include the: Whirlpool Model, Boehm'sFigure Spiral Model, All-at-Once Model, Scrum Model and the Sashimi Model. These variations seek improvements the time, cost and quality of the resulting application software system. These approaches reduce the effects of some assumptions built into Waterfall model.

Results of modeling analysis are many (Figure 2.9). Using many models, a business can: improve organizational communication, reduce the time to respond to customer needs, improve the analytical capabilities of business,
make explicit assumptions and risk factors known, graphically portray critical relationships within the business, and improve decision-making capabilities.

It improves communication within the business since it sets up clear definitions and understanding of the business's terminology. As a result, the business organization can focus

[7] W. C. Burkett, *Process Architecting and Software System Development -- Draft*, University of Southern California, Los Angeles, CA, Spring 1993, 50 p.

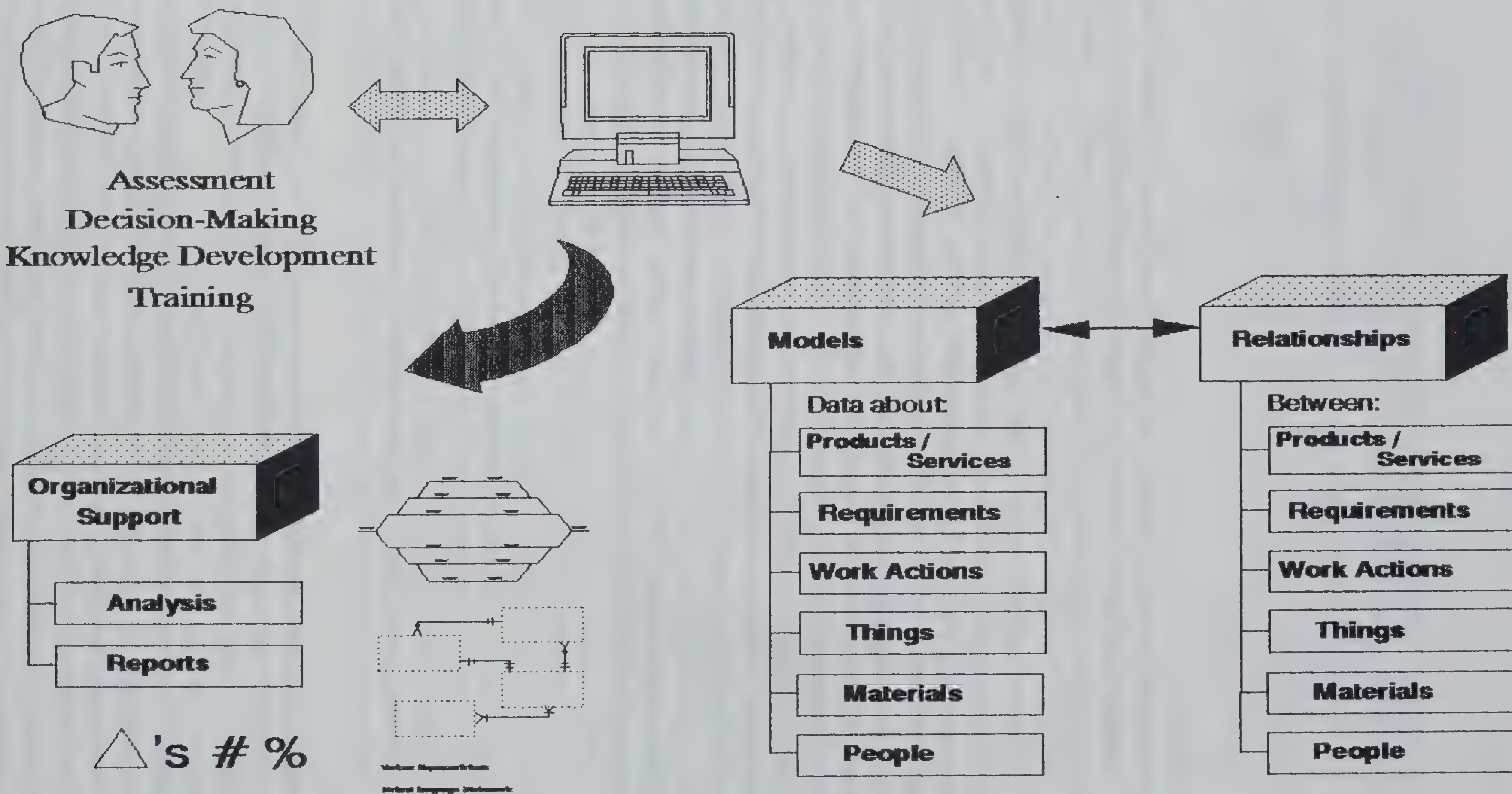
Software Development
Waterfall Model



Are we building the "Right" Software ?

Are we building the Software "Right" ?

Results of Modeling Analysis



on the customers needs and choose the right projects to meet those needs. This reduces time to market since the data, functions, flows, locations, and events are tightly linked becoming more effective. By modeling key processes, the assumptions become explicitly stated and risk assessments are more realistic. This improves the analytical and business assessment capabilities of the business. The various relationships become understood. It influences decision-making in a positive way by structuring stating requirements in a way that produces realistic software solutions.

Realistic software solutions reflect the business' knowledge-base base. To become an adaptive business, the management and employees has to use these models to respond to changing business conditions. This model-based environment supports these practical needs of the business.

The design of information systems considers these fundamental questions:

"What data do we require?"
"How do we process the data to provide useful information?"
"Where and When is the information needed within the business environment?"

These questions deal with the data, actions and locations that bound the design a useful application software system. Another way to view what I/S organizations do is to review an architectural perspective of I/S development work.

An Information Architecture: One definition of architecture is the action or process of designing or constructing: buildings, communities and environmental areas. Contemporarily, it applies to the design of different parts of business and information systems. It is this context that J. A. Zachman describes in Information Systems Architecture - A Framework[8]. He describes some key perspectives and relationships needed for a business to develop, design and build viable, robust, and reliable information systems. He proposes this framework for several reasons. First, the framework provides a clear means for professional communication among I/S professionals regarding various ideas and specifications. Second, the framework shows the need for improving and integrating software application development methodologies and tools. Lastly, the framework fosters credibility and confidence in the wise investment in I/S resources.

Within this architectural framework, Zachman poses several key questions[9]: Who ?, What ?, Where ?, When ?, How ? and Why? The questions are asked from different perspectives. The perspectives he includes are: Planner, Owner, Designer, Builder and Subcontractor.

---

[8] J. A. Zachman, "*A Framework for Information Systems Architecture*", IBM Systems Journal, 26(3), 276-292(1987).

[9] J. F. Sowa and J. A. Zachman, "*Extending and formalizing the framework for Information Systems Architecture*", IBM Systems Journal, 31(3), 590-616(1992).

Each perspective produces a different set of relationships and resulting model, as shown in Figure 2.10. From the perspective of the row Objectives/Scope), a planner generically describes What: things (Entities), How (Functions), Where: (Locations), Who (Organization Units), When(Business Events), Why (Business Goals/Critical Success Factors). With each succeeding perspective (row), there are additional details (specifications) required to express the essence of the design.

To develop effective, robust, application software systems, the I/S organization examines three critical architectural perspectives: computer, software, and network architecture. Each architecture is briefly described here:

Computer architecture is the designing of computer systems to fit the business' needs. It sets up the standards for all software and devices that are connected or run a computer system.

Software architecture is the design of one or more programs which meet a specific businessor industry need. A Network architecture is the (1) design of a communications system, which includes the hardware, software, access methods and protocols used. It specifies whether computers can act independently or are controlled by other computers monitoring the communications network. It also refers to an (2) access method in a Local Area Network (LAN).

Since the word design is common to each definition, this framework provides insights to the design process. For example, computer architecture relates to the design and installation of the Distributed Systems Architecture and Systems Architecture (cells). Network architecture includes several cells: Distributed Systems Architecture, Systems Architecture and Network. Both the DATA and the FUNCTION columns are part of a Software architecture. Thus, each perspective models the elements of design needed to produce a working computer and communications system and the supporting application's software programs it uses.

An integrated system is a collection of distinct elements or parts built into one unit. An integrated application software system combines several applications in one program. Today, we are seeing the packaging of database management, word processing, spreadsheet, business graphic and communications functions. It provides each business person with the capability to use all functions and features in an easy-to-use environment. Looking at the bottom line of the modified chart, a functioning system provides:

1. The data and actions that people communicate with on a timely basis supporting the basic purpose and motivation behind the use of the system.

2. The packaging of data views and software methods that support the business commitments people make linking schedules and knowledge about business objects.

3. Different structures of data and sets of procedures that support people's work in various locations linking any business dependencies and strategies.

Using a framework provides several key perspectives needed to develop integrated systems. But, Zachman makes several points. Each row represents a distinct and unique perspective resulting in a different set of models. Each column is unique but contains

*

Information Systems Architecture - A Framework

Some of the Things & Relationships important to a business include:

	DATA	FUNCTION	COMMUNICATION	PEOPLE	TIME	MOTIVATION
	WHAT	HOW	WHERE	WHO	WHEN	WHY
OBJECTIVES/ SCOPE	List of Business Things	List of Processes (Actions)	List of Business Locations	List of Organizations	List of Events	List of Goals/Strategies
MODEL of the BUSINESS	Information Semantic Objects/ Rules	Process Flow Process/ Products - Services	Business Logistics Location/ Link	Business Organization Organization/ Work Activities	Business Cycle Events/ Schedules	Business Strategy Objectives/ Strategy
MODEL of the INFO. SYS.	Business Data Data Objects/ Relationships	Data Flow Application/ Data View	Distributed C/S System Function/ Link	Work Interfaces Role/ Deliverable	Processing Dependency System/ Duration	Business Decision Criteria/ Options
TECHNOLOGY MODEL	Physical Structure Tables/ Keys	Function Flow Methods/ Format	HW & SW System Design HW-SW/ Function	Work Design Customized/ Job	Control Design Execute/ Assess	Business Design Condition/ Action
DETAILED REPRESENT-ATIONS	Data Design Fields/ Structure	Program Design Language/ Control	Network Design Address / Node	Security Design Role/ Transaction	Timing Duration Wait/ Execution	Knowledge Context Sub-Condition/ History
FUNCTIONING SYSTEM	DATA Views Structure	ACTION Methods Procedures	COMMUNICATION Link Locations	PEOPLE Commitment Roles	TIME Schedule Dependency	MOTIVATION Knowledge Strategies

additional levels of detail needed to support the appropriate perspective from generic to specific. Each cell (row, column) is unique and different representational forms express the relationships contained in the cell. These models are not static representations of the business environment, but are dynamic. Today, a gap exists since most of the models provide static representations where a dynamic representation is needed. Therefore, these models need to be constantly monitored and adapted to the changing needs (requirements, business rules, etc.) which affect business.

This framework contains five rows and has six columns. It results in up to 30 different combinations of models that express the essence of a business and information systems design. An additional 11 model aggregates (row and column summaries) can be developed from the other models. It provides a sense of the common elements needed to deliver an information system for a business.

What are the underlying common elements needed to interrelate each of the different cells? Can we abstract those elements and relationships to produce a more generic or basic model to run the enterprise? In the I/S world, this is called a meta-model.

Zachman's I/S Architecture gives insight into the business and technology issues associated with developing a robust application software system. How can we improve the development process? How can we align, bridge, and deliver the right application software systems for the business's need within the time, quality, and cost constraints? The linkage of these models may help us to bridge some critical business issues associated with software development in manufacturing.

Abstraction

Since a model serves a purpose, a critical idea associated with each model is the generic or specific definition that the model represents. This idea relates to the degree of generality or specific detail known as Abstraction. It is defined as:

The act of considering something as a general quality or characteristic, apart from concrete realities, specific objects, or actual instances[10]. Besides the structuring of definitions, we have to be sure that the higher leveled definitions and characteristics apply to each succeeding lower level definition. In this fashion, the model maintains consistency. Each question: Who, What, Where, When, How and Why shown from Zachman's I/S Architecture provides different perspectives and definitions. With each level of definition, the characteristics and relationships have to be understood.

Another part of the formulation of a model is its scope. For the model, the scope statement describes the purpose, extent or range of application or operation. It is an unambiguous statement that reflects the expectations and operational meanings so all parties understand its purpose and application.

---

[10] J. Stein, p. 6.

Extent and breadth of model:

If the context or scope of a model is broad, it may be an enterprise level model. It means the model represents the things that the enterprise does or acts on. At this level, we have to qualify the type of enterprise model it is. An Enterprise-Information Model shows all the information the enterprise uses. An Enterprise-Business Process Model shows all the major processes the business executes. One can develop a model for a particular business function (e.g., marketing, finance, etc.). On the other hand, smaller scope statements define different types of particular models. Narrow models define particular areas or subjects. Even at this level, the model specifies detail for a particular application or purpose.

Approaches in Modeling: Top-Down: One way is to design is to divide a problem into small understandable parts. This is a black-box approach to design and problem-solving. One can start the design at the highest level of a business (Top). The step-by-step decomposition (divide and conquer) of the problem by specifying additional levels of detail (Down) about each part is known as hierarchical decomposition or stepwise refinement. This pattern is explicitly used in each column of the Information Systems' Architecture. It was explicitly used to formulate the previously shown Functional Model. As a point of comment, "Top-Down is reasonable way to look at various representations dealing with things that are already understood. It may not be a reasonable way of developing, designing, or discovering anything.[11]" Today, this is the major issue. We need creative association and judgement in our designs. I/S are built on logical reasoning, ordering and precision. This logical structuring is dictated by the scientific methods that are the underpinnings of computer system's design.

Bottom-Up: Another approach to design is to start the design from the lowest level of a business (Bottom) and go on to extract the general characteristics of some aggregate (Up). The aggregate is a class of similar objects. This approach to classification is a synthesis process.

Middle-out: Another approach in designing is to start in the Middle of the classification scheme. By examining any inconsistencies in the classification structure, the design can go upward or downward in reviewing the general or specific characteristics (Out).

This summarizes our discussion on modeling from a business perspective. Since the business drives change, these changes have to be incorporated into the business' environment. These are the essential points for this section:

1. Business is a system of interrelated actions and objects.
2. The precise explanation, definition and relationships about each business object affects the workings of wholebusiness.
3. Various models help to provide an insight into the relationships that exist or can exist in a business.
4. A knowledge base of the relationships of who, what, where, when, why and how are critical for improving the business'effectiveness and efficiencies in the 1990s.

---

[11] M. A. Jackson, *Principles of Program Design*, Academic Press, New York, NY(1974), p. 370.

5. The knowledge base consists of sets of models reflecting the difference operating perspectives of the business.

6. The development of I/S software is dependent on this knowledge base since it represents the fundamental requirements to developing quality application software systems.
7. The software development has to change to make use of this knowledge base.
8. The integration of software and systems relates to:
    a. Meaning and context of data
    b. Structuring of the data
    c. Classification of data
    d. Physical storage of the datum and
    e. Availability of datum
    f. Knowledge base of applicable models.

These points provide an insight into the gaps that exist in modeling. These points inhibit rather than promote the alignment of application software systems for manufacturing.

Specific System Analysis & Design Techniques: Developing a software application requires an examination of the same questions posed by Zachman's framework to produce a solution of the business problem being solved. Each technique has a different purpose. The choice of a methodology requires a clear understanding of the results you want to achieve. Each step of a methodology should produce a product valued by your business. These are a few of the commonly used techniques used in the design of a software application.

Data Flow Diagrams show the movement or flow of data from one place--storage location -- (source) to other (sink). The controls, transformations (verb-object pairing) and events that affect the data can also be shown. The transformations are the functions the system performed by the system. Decision Table shows statements of the one or more conditions that exist (Yes, No) and the decisions made based on those conditions. A Dictionary contains the descriptions and definitions of the data about
each piece of datum. For each datum (field, data element or item of data), several other pieces of data are maintained. They include: its name and synonyms,description or purpose, values, format and structure, and relationships to other pieces of data. Extensions to the database that stores the dictionary are described later in the CASE tool section. An Entity-Relationship Diagram shows the relationships that exist between one or more things (Entities, Objects). In some cases, the characteristics (data that describe its characteristics or attributes) are also shown
(ERA). It shows the types of information and their relationships.

Matrices are one way to express different relationships in a simple, easy-to-use format. Matrices show Organization-to-Document, Document-to-Process, Process-to-Information, Information-to-Data and Data-to-BusinessFunction relationships. The choice of the relationship to be examined depends on the analysis needed.

A Structure Chart shows the relationships of actions (Processes, Functions, Tasks, Modules) in a total design. It starts at the top and is decomposed by specifying additional levels of detail.

A State Transition Diagram shows the sequence and control (dynamics) of the way the system behaves over time. A state is the time it takes to do some action (process). The transition is the sequence of the actions between two other processes. The conditions are the rules that cause movement from one state to another. The result of the software development process is an application software system that is another product of your business. By analogy, a methodology is an assembly line to build software products[12]. Each step of a methodology produces an intermediate product(assembly) needed to complete the finished product or part. Each phase of the methodology equates to a work station in an assembly line.

Just as in the development of a new product, parts are designed, classified, inventoried and standardized to achieve a better quality and cost-effective product. This is also the case in the information systems world. If the data, objects, methods, and programs are defined, designed, classified, inventoried and standardized, the same benefits can be achieved. This is the rationale for reuse of software components. If data, software and system components were classified, inventoried, managed and reusable, it could reduce the time needed to produce new and incremental changes for application software.

The following table shows the focus point of each technique compared to the basic questions posed in Zachman's I/S Framework. The bold faced item highlights the technique's primary focus.

There are many variations of the above techniques. Combinations of these techniques are used to describe the total specifications of an application software system.

---

[12] M. Bryce and T. Bryce, *The IRM Revolution: Blueprint for the 21st Century*, M. Bryce & Associates, Inc., Palm Harbor, FL(1988), p. 15.

Name	What	How	Where	When	Who	Why
Data Flow Diagram	**Data &** **Stores**	Action				
Decision Table	Data	**Action**				**Condition**
Dictionary	**Data**					*Purpose*
Entity-Relationship	**Data Relationships**					Rules
Function Decomposition		**Action**				
Matrices (Two or 3 Dimensions)	Data	Function	Location	Time	Organization	Goal Mission
Structure Chart	Data	**Action**				
State Transition Diagram		**Action**		Event		Condition

Table 2.3

Each technique uses different notations and styles to convey the specifications and context of the system being designed. These techniques include:

Entity-Relationship Modeling (ER) - Developed by Peter Chen in 1976, this method describes and graphically shows how information (entities) and relationships describe requirements for systems development[13]

Information Engineering (IE) - An interlocking set of formal techniques for the planning, analysis, design and construction of information systems. It is applied

[13]Chen, P. P-S., *The Entity-Relationship Model-Toward a Unified View of Data*, ACM Transactions on Database Systems, Vol. 1 No. 1 (March 1976), pp. 9-36.

on an enterprise-wide basis or across a major sector of an enterprise[14]. The automated collection, documentation and reporting of the data collected in each application software development phase is part of the CASE tool environment.

Integrated Computer Aided Manufacturing DEFinition Language (IDEF) - Integrated Computer-Aided Manufacturing program (ICAM)[15] program was set up to improve the productivity of manufacturing information systems through the systematic application of computer technologies for the aerospace industry. ICAM Definition Methods provided a means for computer analysts and manufacturing professionals to design, discuss, and record the requirements of the manufacturing system. There are three ICAM methods:

Name of Method	**Type of Model**	**Focus`**
IDEF-0	Function	Action
IDEF-1	Information	Data - Relationships
IDEF-2	Dynamics	Event

Two levels of models may be developed for each method. One describes the current or "AS IS" state while another describes the future or "TO BE" environment. Each model includes a Glossary of Terminology and a depiction of the: Material Flow, Manufacturing Functions, Information and Information Flow.

Nijssen's Information Analysis Methodology (NIAM)

NIAM focuses on data as the center point of design by assuming sentences and propositions are facts stored in databases. Each data structure (fact type, entity) is affected by both static and dynamic restrictions.

Object-Oriented Modeling (OO)
Object-Oriented Modeling deals with capturing the design of the data about the object (entity) and the behavior (action) of the object in the real world.

Product Data Exchange using Step (PDES). PDES is developing a complete, unambiguous, computer definition of the physical and functional characteristics of a product throughout its life cycle.

Structured Analysis and Design Technique (SADT). SADT defines system requirements using a series of procedures decomposing the requirements into the: a Function or Activity Model in an actigram and Information and control elements in a datagram.

---

[14]Martin, James, Information Engineering: Book II Planning and Analysis, p.vii

[15]Integrated Computer-Aided Manufacturing (ICAM) Architecture, Wright-Patterson Air Force Base, Ohio 45433, SofTech, Inc., 1993, p.1-5.

These briefly describe techniques available for use. I would suggest the reader review Software Design Techniques by Peter Freeman and Anthony Wasserman[16] for addition insight into these and other techniques.

Key principles for Information Management

In their discussion of Information Resource Management, Byrce's laws provide an interesting background on contemporary techniques for designing information systems[17]. A major issue is that systems development is a problem-solving activity. Therefore, the steps, procedures and infrastructural elements the business places around this process have a tremendous bearing on the software product produced. Thus, Bryce's laws are very insightful. For example:

All organizations have a data base; some are managed, most are not.

A critical element in today's business is knowing what information the business has and how it can be most effectively used. This amounts to knowing the data. Specifically, it means knowledge of the data -- its meaning, definition, synonyms, formats, currency and relationships -- is fundamental for the business' operations. These semantics and any tools used to manage this asset are important to the business's success.

CASE Tools to support Modeling: The process of designing, coding, installing, and maintaining software applications is complex and expensive. This is driving the CASE industry. Computer-Aided Systems Engineering (CASE) technologies are software programs that automate many techniques discussed. There are hundreds of CASE products available offering many different capabilities and functions[18]. One classification commonly used for CASE functions is shown in Figure 2.11.

Upper CASE	Documents business requirements
Middle CASE	Captures elements of design and analysis of application software
Lower CASE	Generates, tests and maintains software instructions for a specific software and hardware environment.

Since CASE tools address many processes and functions, there is no one product that satisfies all the functions.

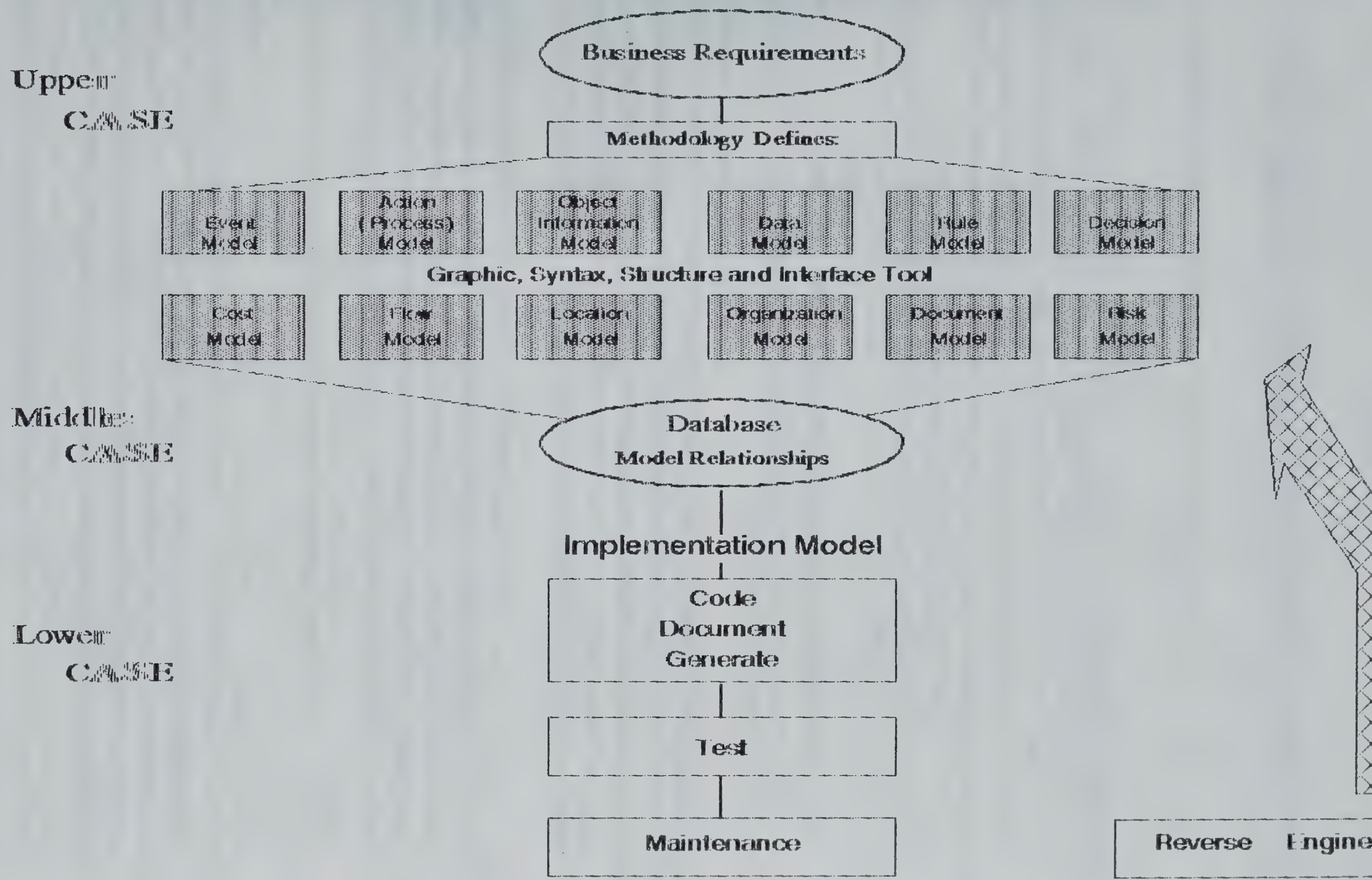
---

[16]P. Freenam and A. I. Wasserman, *Software Design Techniques (Fourth Edition)*, IEEE Computer Society Press, Los Alamitos, CA (1993).

[17] M. Bryce and T. Bryce, op. cit., p. 244-248.

[18]G. Forte and K. McCulley, ed., *CASE Outlook: Guide to Products and Services*, CASE Consulting Group, Lake Oswego, OR, 1991.

Computer-Aided Systems Engineering (CASE)



Since there is no integrated set of tools, the manufacturing business managers have to choose the set of tools which best fits their business' application software development process. One has to consider tools for: Project management and risk assessment (methodology), data (information-data Models), process (process analysis, flow models), database (repository), generation (automatically generate specific software instructions to fit hardware, software, database, communication environments), test and evaluation of resulting code (testing, debugging and maintenance) and reverse engineering of existing software to place them into the database of the CASE tool.

Really, the choice is about how you decide to manage the process of software development. There are risks associated with using CASE. You need specifically to have:

1. Management support for a CASE environment.
2. An Organizational willingness to use CASE.
3. A willingness to investment in training and computer tools.
4. The ability to provide integration across software tools.

The absence of any of these elements can dramatically affect your use of the tools. When using CASE tools, management has to actively participate and understand that the requirements are models that reflect the way they do business. This means management has to agree, understand, set standards, allocate resource and proactively use the adopted methodology. It is both a cultural and technological change in the way they do business. Business and I/S organizations have to adopt the right CASE tools to fit their needs. This means agreeing on a methodology and associated processes before the tools are chosen. Otherwise, the use of the tools is ineffective. Once the methodology, process and procedures are set up, the business has to continually invest in training and tools necessary to become more effective. Lastly, the business has to invest in other services to provide the integration required across the CASE tools chosen. If any of these elements are missing, the probabilities of its successful adoption are reduced.

In the CASE environment, a dictionary -- sometimes called the database, knowledge-base, project manager, encyclopedia or repository -- stores the data about the systems development process. It stores the data and relationships captured by the individual techniques the CASE software manufacturer has packaged together. The dictionary uses many software packages to support these functions, such as: configuration, editing, word processing and report generation, measurement of programming functions, project management, verification and validation, and simulation of model changes. The CASE environment focuses on generating application software systems from different types of dictionary based systems. The trend of the environment is correct, but, there are extensive unmet needs. They are:

1. How to deliver a model-based management system of the manufacturing business' operation?

2. How to distribute knowledge-based elements throughout the business effectively to do object, relationship, and model analysis?
3. How to evaluate terminology (Semantics) used in the business?
4. How to classify and structure data dynamically with the associated terminology?
5. How to access relevant data across the manufacturing business from this knowledge-base modeling environment?
6. What additional business analysis tools should be incorporated with knowledge-base environment to help business users become more effective?
7. How can queries and reporting of model based facts be simple and easy-to-use?
8. How can the presentational style be personalized to the person's natural means of communicating?

3. Modeling Gaps

Several major gaps which exist in the development of models today. They are categorized under these major areas with secondary items outlined under each. An explanation of each gap is given in this section.

One major gap is that there is not an easy means to portray the business relationships and the information about those relationships across the business. There are several aspects to this gap.

1. A major gap today is improving business performance without thesupporting business analysis tools and framework to link to the systems development process with it. This is because most models are developed independently and are not easily accessable for the business' use.

2. We lack tools and techniques to expose the assumptions and relationships (rules) in existing application software systems. This adversely affects the business' performance.

3. Without a collection system to gather employee knowledge for the integration of models, a business loses opportunities to focus in on the right business solutions.

4. We lack automated mapping tools to easily shift from one graphical style and syntax into another methodology style and syntax.

5. A gap exists in providing tools and techiques used in business analysis within the CASE tool environment.

6. We lack a business database or repository for the integration of different models.

Another major gap is there is no easy means to easily assimilate various modeing methodologies with these new technologies.

1. The absence of formats for CASE interchange of existing and proposed standards makes the assessment of these models is very difficult.

2. There is no means to use native language interface into the knowledge-base of models.

The alignment of strategic business and technology planning systems is another area where a gap exists.

1. A gap exists since a linkage has not been made to incorporate creative thinking techniques and tools used in business planning with those used in the systems development process.

2. Another gaps relates to share data captured in different planning and modeling tools. There is no tool to easily do this which results in a manufacturing business leaving itself open to errors in planning.

3. A gap exists since most of the models provide static representations dynamic representations are needed.

Another major gap is the absorption of new technologies into the manufacturing enterprise's culture.

1. A gap exists due to a lack of an electronic standard glossary of manufacturing terminology to introduce into the modeling processes.

2. Today's independent modeling development efforts produce conflicting terminology, meaning and synonyms and classifications that cannot be easily cross-referenced.

3. The availability of generic libraries of manufacturing objects is an existing gap to migrate into Object-Oriented technologies.

4. The lack of a software application to generate models into various heterogeneous Database Management Systems (DBMS) affect the time it takes to deliver the application software.

Because of the competitive environment, we need to understand the multifunctional and multidimensional aspects of a manufacturing business. This is most easily done when knowledge-based models are available for the business' use and are maintained in some database. The users of these models include: executive and general management; empowered team decision-makers, horizontal process managers and operational personnel, information developers, planners, application developers and every person who is part of the operational business even subcontractors. This point was made by Kowalkowski[19] that we lack a business database or repository for the integration of different models.

Assuming the knowledge-based model relationships Figure 2.12 involves some categories shown here, the use of the database becomes a critical component of the business' infrastructure. Similar categories and relationships were part of a system developed for a DARPA research project[20].

People who use this knowledge-base can examine relationships at a strategic business level or at an operational level without any loss of information. The contents of the knowledge-base link various management approaches and techniques. It provides a means to do strategic and technology planning, process improvements and any analyses required to develop application software. The database relationships are key elements of the interprocess management system needed to support information resource management[21]. It provides a means to answer the questions: What, How, Where, Who, When, and Why which Zachman asked in his framework for an Information Systems Architecture. The database becomes a vital link for improving business performance. Organizations have to develop their own database from off-the-shelf software that do not share models or their data easily. A major gap today is improving business performance without the supporting business analysis tools and framework to link to the systems development process with it. This is because most models are developed independently and are not easily accessable for the business' use.

Today, inter-discipline teams working together to drive new levels of business performance. An example of one strategy is Design for Manufacturability (DFM). A

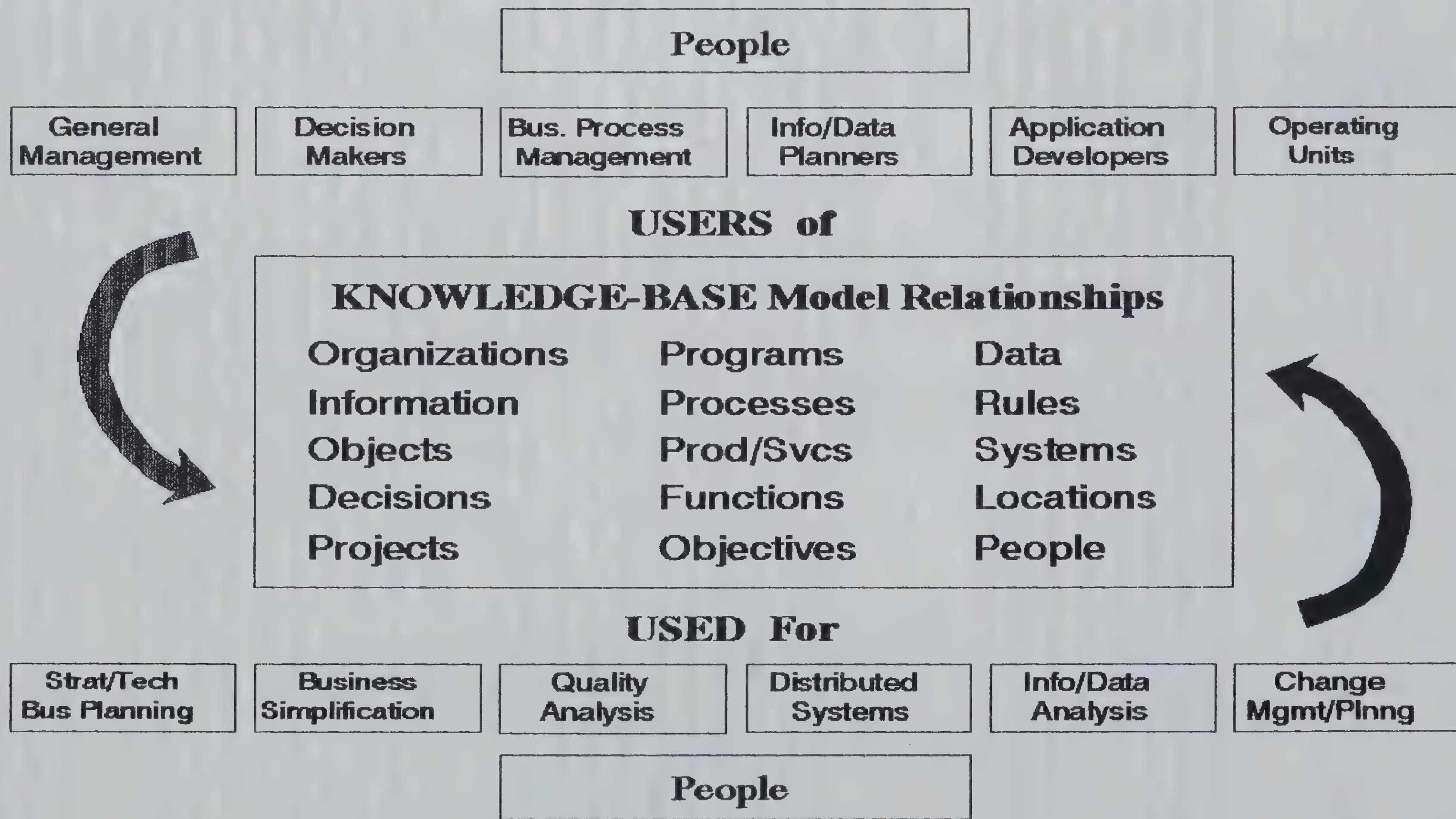
---

[19] F. F. Kowalkowski, *The Dynamics of Models and the Impact on Data*, Guide 86 Meeting, March(1994), Los Angeles, CA.

[20] A. Famili, D. S. Nau and S. H. Kim, *Artificial Intelligence Application in Manufacturing*, The MIT Press, Cambridge, MA(1992), p. 349-383.

[21] J. T. Metcalfe, "*Information Resource Management - Data Ownership, Organization, Roles and Responsibilities*", IBM Business Systems-Internal Report, 1-49 (1988).

Use Of Business Knowledge-base Models



multifunctional team develops a new product design. It examines industry and proprietary standards. It develops complete product and manufacturing specifications. The team does competitive analysis and comparisons, completes product definition, and even makes changes affecting the business' organization and its reward system. Error rates, cost savings, and cycle time reductions range from 25% to 87% [22]. Similar strategies require a broader perspective of the business' operation to examine multiple alternatives.

Today's decision-making, within most larger organizations, is conflict-prone, suboptimal, and slow. This is due to the highly interdependent and multi-leveled decisions that must be made across larger vertical (silo) organizations. Many of today's operational application software systems are built to support vertical organizations. Therefore, the structure of the data within each application is specifically for use by that organization. This also impedes decision-making since the data to produce information cannot be easily used across these application software systems. Critical business rules and assumptions are in these applications. We lack the techniques to expose the assumptions and relationships (rules) in existing application software systems. Lacking this knowledge decision-making under these conditions affects the time-to-market cycle significantly and is a gap for manufacturering businesses.

Besides the inability to expose these assumptions, data, and relationships, businesses are losing knowledge. The knowledge is being lost comes from experienced employees who leave the business. These individuals have the business knowledge about these applications. This knowledge is an untapped resource. Teaming has become a viable management approach because it captures some of that knowledge. As a result, new models are built independently of old models. Since new models cannot be easily evaluated and compared with underlying assumptions, rules, ideas and definitions with old models, it causes organizational chaos. Not having the evaluation techniques, the changes caused with downsizing and rightsizing, may be done for the wrong reason. These organizational changes require the business to reexamine and recreate or drastically modify its existing application software systems. Without a collection system to gather employee knowledge for the integration of models, a business loses opportunities to focus in on the right business solutions.

A Planning View of Models

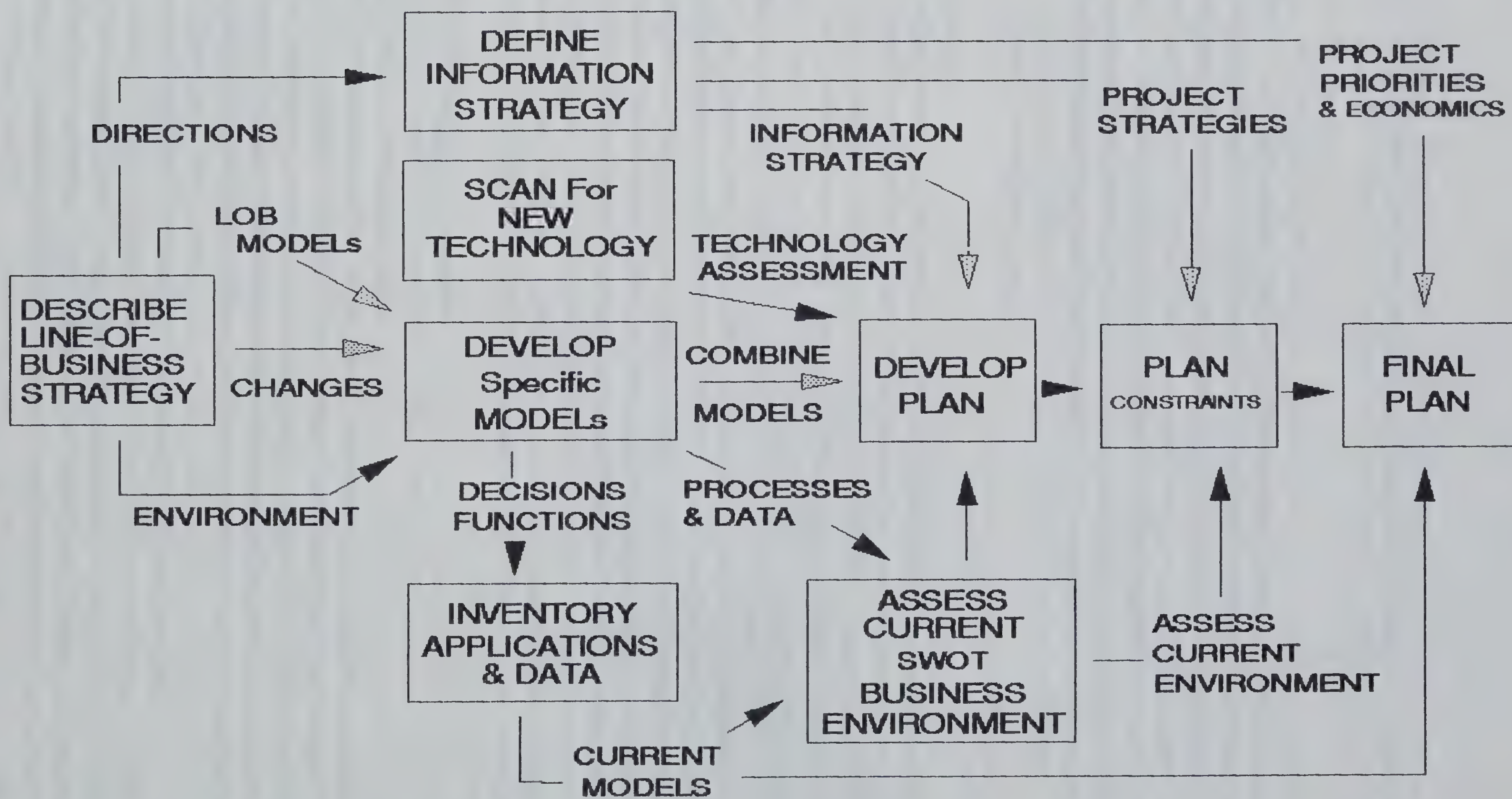
From a planning perspective, the linkage of models becomes a critical part of meeting the manufacturing business's objectives and goals. Although there are many organizational perspectives, scanning for technology solutions, aligning them with business objectives and assessing their potential is required. Here is a typical planning process (Figure 2.13).

If your core business covers multiple lines of business, each business unit develops a particular strategy. The strategy outlines the business' environment, its fundamental approach and directions, and its key assumptions. The use of different models help to develop the strategy. For example, using a combination of financial, organizational, process, event,

---

[22] H. J. Steudel and P. Desruelle, *Manufacturing in the Nineties: How to Become a Mean, Lean, World-Class Competitor,* Van Nostrand Reinhold, New York, NY(1992), p. 83.

A Process for BUSINESS PLANNING



functional, decision, information, document and object models, management creates its Information and business strategies. They highlight the gaps. Specific plans describe how the business intends to close the gaps. Assessments of new technologies and their implications are part of specific projects. Risk assessment and ranking of each project sets up the final plan and course of action. This approach provides a way to align both your business andTechnology strategies[23].

Since business planning is not a smooth work flow, models are built independently of each other. Examining different journals provides you with an insight on the degree of misalignment that exists between the development of the business and information strategies. Another gap exists since there is no easy means to capture and share data linked across modeling tools. This can cause major discrepancies in the planning process since the data cannot be easily shared and accurately examined.

Use of Creative Problem-Solving Techniques

Creative problem-solving techniques help improve business' performance by coming up with creative ideas to solve a problem. Tapping the creative ideas of customers, partners, associates, and employees helps to focus the problem being examined. New application software packages help to collect the personal insights and assist in the group dynamics to come up with the new ideas and solutions. Packages, like Quality Functional Deployment, Group Decision-Support, and Work flow Analysis systems, help to analyze and focus on multidimensional aspects of customer requirements, unstructured problem analysis, and improvements in the flow of work across people, systems, organizations, and cultures.

Considering the virtual corporation, outsourcing and the number of business relationships that exist, these creative techniques have to be linked to the modeling database. A gap exists since a linkage has not been made to incorporate creative thinking techniques and tools used in strategic business planning with those used in the systems development process. The results of these approaches drive changes to product design, manufacturing operations and management, organizational and technology changes not necessarily to the software development process.

Issues in the Software Development Process

A major cornerstone of software development is the scientific principle that divides something into more refined parts. The creative nature of associative thinking processes has not been a major contributor in the software development process. Techniques used for problem-solving which drive multidirectional rather than linear thinking is important. Developing keyword indexes of important characteristics by mixing and matching them in matrices is a way to produce new idea combinations. This helps to identify new characteristics and features that are valuable tools in creative problem solving. These techniques are not

---

[23] G. W. Laware, "Strategic Business Planning: Aligning Business Goals with Technology, Information Systems Management, 8(4), 44-49(1991).

easily incorporated into the lateral thinking process used in system development. The absence of creative thinking techniques and tools in the systems development processes inhibits opportunity analysis and assessment. This gap can improve ideas and business solutions that are applicable to the systems development process. They should be incorporated into business, information, and technological modeling processes to answer the fundamental questions of What, How, Where, When, Who and Why.

A lack of a common manufacturing vocabulary makes it difficult to solve problems. Specifically, this affects the management and interchange of data supporting the business. This is shown in the following way. How did you distinguish the meaning of the term? Is the use of the term consistent within your business or across your industry? Without a set of common terms and their meaning, the communication and interchange of data about the business become fuzzy. This drives a need for common terminology, ideas and characteristics relating various business objects. Glossaries and portfolios of standard manufacturing objects drive the effort to define manufacturing product data. Across a manufacturing organization, there are definitions of many objects we use. These definitions are not readily available in a glossary or portfolio of objects. Let's take, for example, an accepted APICS[24] definition of a Bill of Material (BOM) which is:

Bill of Material (BOM)

A listing of all the subassemblies, intermediates, parts, and raw material(s) that go into the parent assembly showing quantity of each part required to make an assembly.

It is the relationships of parts used in a product. There are a variety of types and display formats used for a Bill-of-Material depending on the intended business purpose. Some of them include:

Costed	**Indented**
Matrix	**Modular**
Phantom	**Planning**
Single-Level	**Transient**

Similar to the mapping done for EDI, a manufacturing business needs to review and choose the appropriate terminology, its characteristics, synonyms and definitions that appropriately fit their organization. Today's independent modeling development efforts produce conflicting terminology, meaning and synonyms and classifications that cannot be easily cross-referenced.

The availability of the database and its terminology would result in a savings both in time and resources. Each business object could be accessed, examined and changed to suit a particular business purpose. Each business object, like the Bill of Material illustration, should be available for classification, definition and customization to fit one or more of the business'

[24] *APICS Dictionary*, 4th Edition, American Production and Inventory Control Society, Washington, D.C.(1980).

needs. A gap exists due to a lack of an electronic standard glossary of manufacturing terminology to introduce into the modeling processes. Using the database as a source of a business's terminology and relationships would speed up applications software systems development time.

Need for additional tools in the Development Process

Such a knowledge base database needs several tools to provide a robust environment to help the manufacturing business and its application software development process. Several types of tools can be combined using:

A Native Language
Many Graphical Styles and Syntaxes
An Image Modeling System
A Classification and Indexing System
Simulations of Business Flows
A Financial and Risk Assessment
A Generator to support heterogeneous DBMS linkages

Here, the point is human communications deal with the terms, meaning and structure of messages between people. Whether the messages comes in person, over telephone or through a software application, the models deal with the elements shown here.
A native language application software system can explore, extract, change or construct new models in the database. Using native languages such as English, French, German, etc., each individual can use one or more sentences to examine various model relationships. Using this language tool that understands the structure, syntax and basic concepts of the native language, a person can probe the database to refine and scope the problem[25]. Once the analysis is complete, the native language creates new models or modifies existing models. Each sentence can be either typed or spoken directly into the knowledge base (Figure 2.14).

If your organization serves many marketplaces, certain models may have to be changed from one style of representation to another. If your business markets to both the commercial and military customers, the markets may have different standards and requirements. Military customers require that certain graphic standards and methodologies be used for application software development projects. This means any work done has to meet these standards. If a graphical and methodological mapping application software system existed, models could be easily translated from one standard (i.e., Action Diagrams, Entity-Relationship) to another (i.e., IDEF0 and IDEF1x). Since various methodologies have their own particular graphical syntax and style, another gap exits. We lack the capability to automatically mapping of one methodology into another's graphical styles and syntax.

---

[25] J. M. Ginsparg, *A robust Portable Natural Language Data Base Interface*, Proceedings on Conference on Applied Natural Language Processing, Santa Monica, CA, February, 1983, p. 25-30.

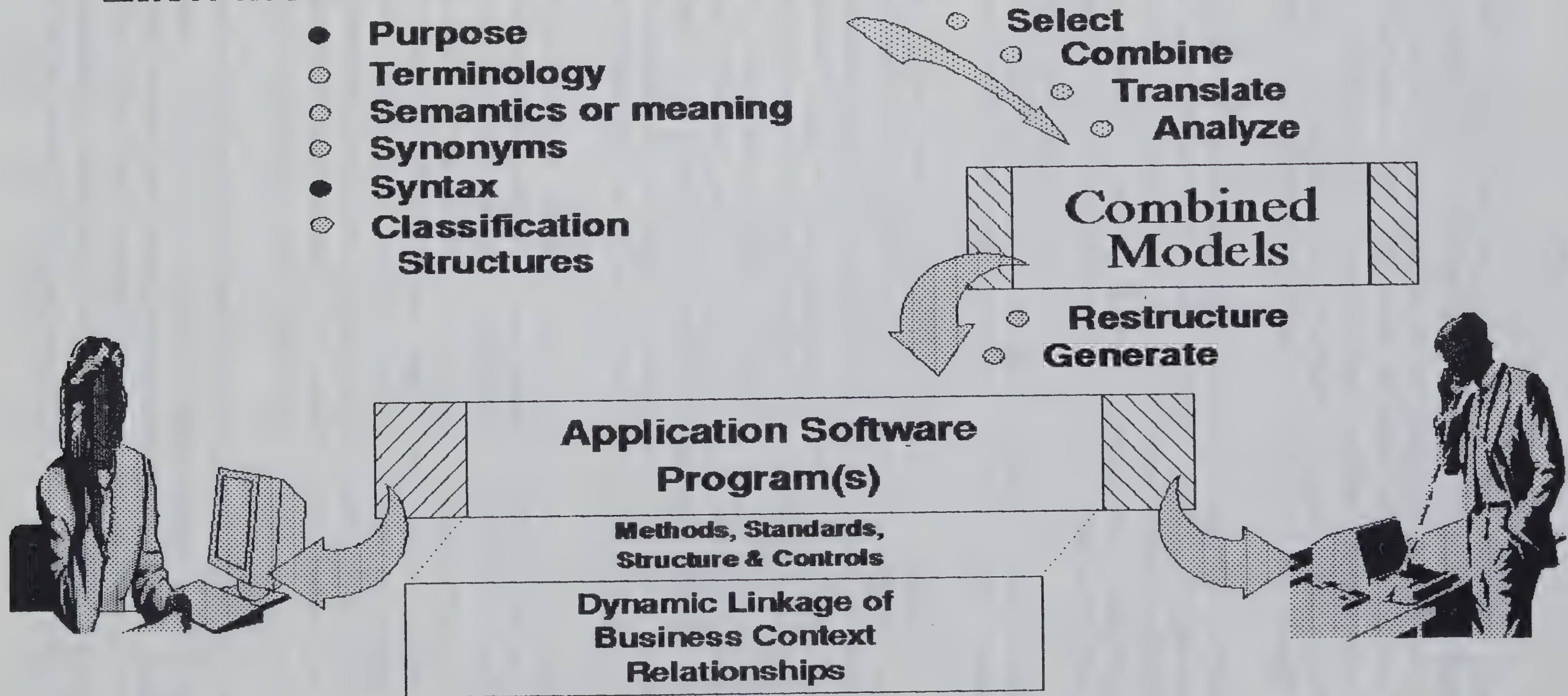
Business Context

Use Native Language to navigate across the Knowledge-Base

Model Relationships for various Business Perspectives

Each Model has

- **Purpose**
- **Terminology**
- **Semantics or meaning**
- **Synonyms**
- **Syntax**
- **Classification Structures**



Today, pictures are part of various application software packages. Photographic images (cars, people, documents) are viewed by the business professional through the software application. This trend will effect the object modeling discipline. An image modeling system will pictorially represent many business objects used in a manufacturing business. Graphic illustrations of tangible objects -- such as a screw, hammer, milling machine, robot, forklift truck, etc. -- provide an easy way for people to identify objects and relationships of the system being modeled.

With multimedia and virtual reality computer environments, graphical images of business objects (2 or 3 dimensional representations) are common. A new process flow could be simulated. Images will more effectively show the relationships in each model. On the other hand, abstract ideas are more difficult to show. A group of manufacturers should develop some standards to define the images associated with abstract concepts. As the knowledge base database contains graphic linkages to specific business objects and its relationships, the image modeling application system would provide realistic and effective illustration of the objects and their relationships. Without some mapping of graphical images to key manufacturing objects for model purposes, the development of resulting applications software will be slow.

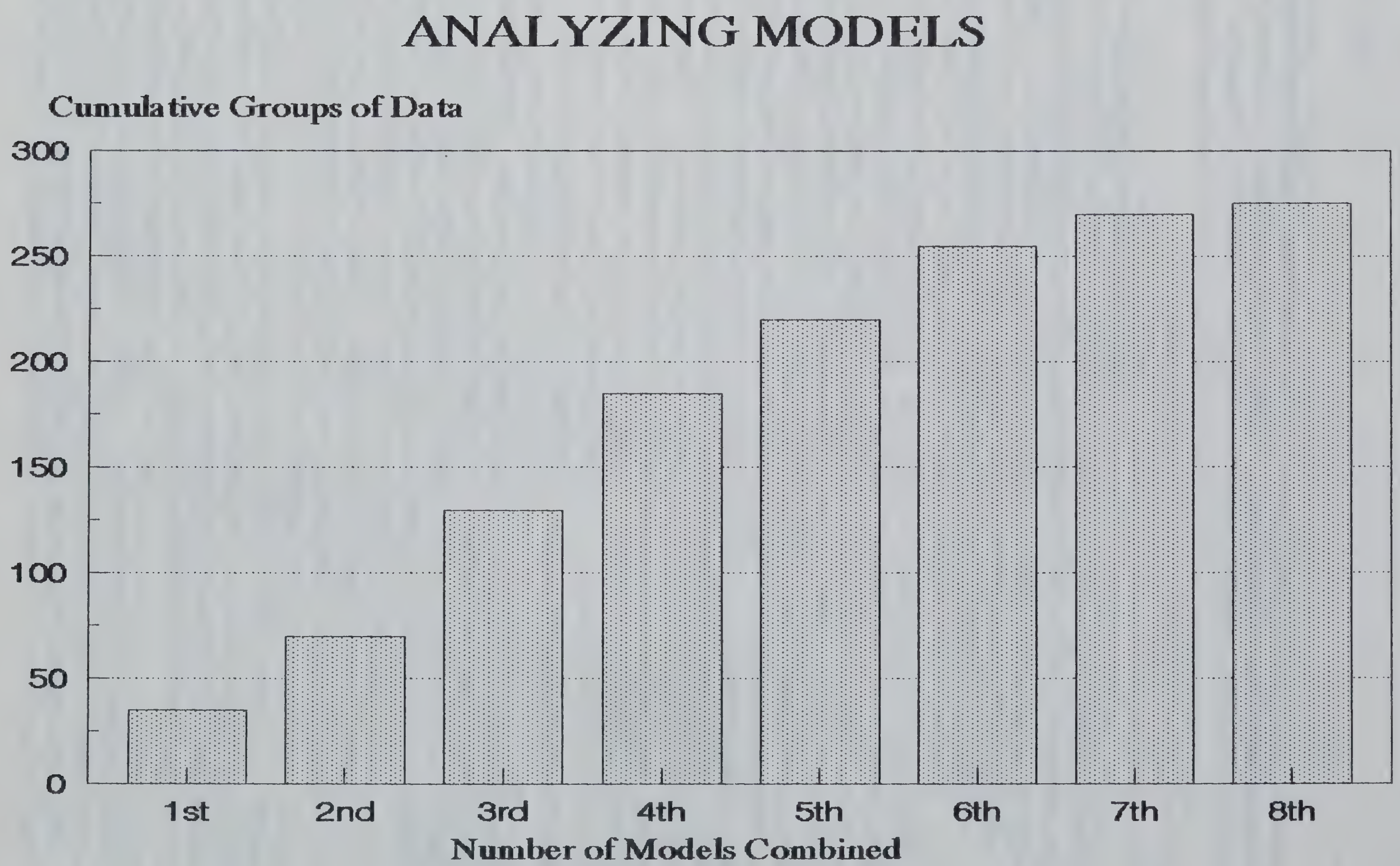
A classification and indexing are part of the everyday manufacturing operation (i.e., parts groups). Just as these groupings allow people to cross-reference and assess the impact of different sets of parts, models can be indexed and classified. A manufacturer deals with both external and internal models. Since standards are defined by different organizations, the choice of the right standard is an important business decision.

Here's an example of external models that describe existing industry standards. EDI and Product Data Defintion using STEP are externally defined standards that affect your business. By capturing the standard (its data, definition and relationships) in the model database, the standard is available for analysis and assessment purposes. Manually, people do an analysis (usually a linguistic comparison) of the standard against your operating environment.

If the models were captured in your database, a quick assessment of the impact of the standards on your business' operational environment would be achieved. This assessment includes the risk and financial implications on your business.

This assessment points to those projects most critical to your operations success. The publication of a CASE tool interchange format of the standard model would speed up the evaluation process. The absence of formats for CASE interchange of existing and proposed standards makes the assessment of these models is very difficult. This gap requires each manufacturing business to capture these standards into its own CASE software development environment.

Models can provide a means to understand aspects of Design for Manufacturing (DFM) and Design for Assembly (DFA) approaches. The standardization of names and coding is critical for the deployment of these strategies. Indexing and classifying business objects such as process plans, feature combinations, parts, documents and others leads to various economies using group technologies. The same principle applies to the classification of data across your business as shown here. When combining different groups of data across several models, the number of unique data groups and datum level off (Figure 2.15). This suggests a stable level



ANALYZING MODELS
Cumulative Groups of Data
300
250
200
150
100
50
0
1st
2nd
3rd
4th
5th
6th
7th
8th
Number of Models Combined

of data supporting the business. New data groupings are added only when there is a change to a line of business. Therefore, new business data must be inventoried, defined and assessed. Once taken, such an inventory is not as dynamic but is quite stable. This is similar to the analysis done for the classification of features[26]. The classification and indexing of data are a key element of various operational activities and flows.

There are various flows that exist in a business. The characterization of the flow is critical in planning its effective use. The simulation of the flows measures various time, cost and resource usages. Using Program Evaluation and Review Technique (PERT) and Critical Path Methodology (CPM), what-if projections evaluate both time and resources. In the I/S world, we use Data Flow Diagrams (DFD) to represent the movement of data to or from a process. Other network analysis techniques show the relationships of actions, time and use of resources to find an optimum path for communications and database access systems. Simulations that characterize volume, time and costs should be a part of the analysis supporting a DFD or Process Analysis. The delay in measuring, comparing and simulating the time, cost and resource characteristics of process and data flow design increases the probability of an erroneous design. Closing this gap will provide a better understanding of the actual use of the application. In this way, each business object can be judged by its contribution to the business's financial objectives.

One factor driving a manufacturing business is profit. A financial and risk evaluation of each business choice is part of the business' operation. This evaluation applies to making or purchasing any application software system. Usually, a financial (Break-even or Internal Rate of Return) analysis evaluates the impact of the system on the business's operations. Since each object described in the model contributes to the business' performance, any financial factors attributed in the financial model should be part of the knowledge base relationships. Today, these financial and subjective risk factors are not directly linked to the business objects. Direct measurement is not available making the historical measurement difficult. An evaluation methodology, like Information Economics, combines a return-on-investment approach with a project-scoring technique to provide a more realistic means to justify information technology. It connects three ideas: value linking, value acceleration and value restructuring. Value linking connects the effects of information technology with strategic business planning results, which are measured by increased revenue, decreased costs, or accelerated growth. Value acceleration is the ability to attain one-time benefits and reduced costs earlier using information technology than without that technology. Value restructuring examines the productivity of personnel effected by the influx of the technology on the manufacturing business. The approach examines the problem of how to measure and justify the necessary investments in information technology. Without addressing the realistic measurement of various models components, the implications of introducing new technologies and software on the manufacturing business, are difficult.

---

[26] A. Houtzeel, "*Computer-Aided Process Planning and Group Technology*", Advanced Manufacturing Technology, U.S. Department of Commerce-National Institute of Standards and Technology, Bethesda, MA(1992), p. 25-40.

Today, models exist in various tools (CASE, Spreadsheets, Word Processor). Most of the tools have proprietary development methodologies. The lack of a software application to generate models into various heterogeneous Database Management Systems (DBMS) affect the time it takes to deliver the application software. This inhibits the use of the application across many environments due to navigation and database management system structures. The tool would generate both the Data Definition and Manipulation Language (DDL/DML) interfaces necessary to support the model's use across these various heterogeneous database management systems. This is the reason productivity improvements can occur with the use of reusable software components linked to business objects.

One key advantage of object-oriented software development is the packaging of objects that have similar characteristics and behavior into an object class. The behavior is a set of actions defining the interactions, functions, responses and rules that relate to the object. The object includes all of the data related to the object and rules about the values of the data. These elements are combined into a reusable software package. The availability of generic libraries of manufacturing objects is an existing gap. Beyond the availability of a generic manufacturing object library, the relationships, methods, rules, values, definitions should be independent of a particular technology environment. A gap exists because class libraries are developed and restricted to particular software and hardware environments.

4. Conclusions

A manufacturing business has to align its objectives with those of its customers. Simultaneously, it must remain competitive by delivering valued products and services. Using multiple sets of models is a key part of planning, organizing, directing and controlling the way the business operates in its environment. These models aid in decision-making by changing the way the business operates. These same models drive changes based on the requirements of the business. Models become the operative plans that effect information technologies and application software development process. Because of their importance, these models should be incorporated into the business' knowledge-base for application software requirements generation.

Today's business performance and improvements are affected by technology. The absorption of that technology into the culture requires a significant amount of capital investment, retraining, and education before it is fully absorbed into the business. Using Computer-Aided Design(CAD), Computerized Numerical Control (CNC), Computer-Aided Process Planning (CAPP) and Computer-Aided Manufacturing (CAM), Materials Requirements Planning (MRP), and Manufacturing Resource Planning (MRPII) drive significant changes to operations of the business. Integrating these applications means managing the business context and their interrelationships.

We have discussed the essential tradeoffs in the development of information technology systems in manufacturing based on quality, cost, and time. Technology is becoming an essential ingredient in the infrastructure of running a business and application software solutions are critical to the business' success. Since the application software development

process emulates the business's actions by providing data access, business rule enforcement and in some sense, becomes the context of the business' operations. Therefore, models help to provide the context of the business' operations and provide the basis for technology implementations.

Quantum leaps in productivity in the 1990s and into the next century are critical for manufacturers vitality. The use of an interactive, interrelated, distributed generative model management process for the manufacturing is a key element to achieve that productivity. Since models define the changing specifications needed by the business to remain competitive, application software development requirements will be driven from this knowledge-base of business relationships. Only when some of the gaps are addressed will manufacturing businesses be able to achieve the growth and vitality needed for the 20th century.

About the Author: Gilbert Laware has worked in information systems for over 25 years. He is presently a consultant specializing in business planning, CASE Development and information management. Mr. Laware graduated with a Bachelor's Degree in Economics from Siena College. He received a Master's Degree in Management from Fairleigh Dickinson University and a Master's in Management Science from Iona College. Mr. Laware has spent 15 of his 25 years in the information systems arena. This included jobs such as Manager of Business Data, Manager of Data Architecture and Vice President of Information Resource Management. The remaining 10 years were spent in business planning and strategy development. In the last 5 years, Mr. Laware has consulted with in the area of modeling concentrating in the manufacturing industry. He has spoken at IGES/PDES conferences, Data Dictionary Symposium, DAMA Conferences, SHARE and GUIDE meetings.

Mr. Laware is an adjunct faculty member of Iona College's -- Hagan Graduate School of Business and was the Vice President of Operations for the Data Administration Management Association. Association.

Additional Resources

Books

Designing Quality Databases with IDEF1x Information Models by T. A. Bruce, Publisher: Dorset House Publishing

Object-Oriented Systems Analysis by D. W. Embley, B. D. Kurtz, and S. N. Woodfield Publisher: Yourdon Press, Inc.

An Introduction to Information Engineering: From Strategic Planning to Information Systems by Clive Finkelstein Publisher: Addison-Wesley

Software Design Techniques by P. Freeman and A. I. Wasserman Publisher: IEEE Computer Society Press

Information Engineering -- Series by James Martin Publisher: Prentice-Hall, Inc.

Conceptual Schema and Relational Database Design By G. M. Nijssen and T. A. Halpin Publisher: Prentice-Hall, Inc.

Handbook of Data Management by Barbara Von Halle and David Kull, Editors Publisher: Auerbach Publishers

Magazines

Database Programming & Design
Miller Freeman Publications
600 Harrison Street
San Francisco, CA 94107

Manufacturing Systems
Hitchcock Publishing Company
191 South Gary Avenue
Carol Stream, IL 60188-2292

Professional Societies

DAMA (Data Administration Management Association)

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017-2394

CASA/SME (Computer and Automated Systems Association)
1 SME Drive
P.O. Box 930
Dearborn, MI 48121-0930

Consultants & Educators

D. Appleton Company (DACOM)
225 S. Sepulveda Blvd., Suite 300
Manhattan Beach, CA 90266

Data Administration, Inc.
301 North Hamilton St.
Building B, Suite 203
Princeton, NJ 08540

Performance Development Corporation (PDC)
45 Montgomery Knoll, CN 861
Princeton, NJ 08542

Zachman International, Inc.
2222 Foothill
La Canada, CA 91011

Key Vendors

CASE Tools
Asymetrix (InfoModeler)
Bachman Systems (Bachman Analyst)
Information Engineering Systems Corporation (IE: Advantage)
Intersolv, Inc. (Excelerator)
LogicWorks (ERwin)
Knowledgeware, Inc. (IEW)
Oracle Corporation (CASE)
R&O (Rochade)
Texas Instruments (IEF)

Database

Borland (dBase, Paradox)
IBM Corporation (DB2)
Microsoft Corporation (Access, FoxPro)
Oracle Corporation (Oracle)
Sysbase, Inc. (Sysbase)
Tandem, Inc. (Tandem)

Chapter 3
Manufacturing Enterprise Functional Architecture

Richard W. McHard
SMS Software Management Incorporated
750 The City Drive South, Suite 420
Orange, California 92668
714/740-1113

Abstract

Rapidly changing technology has created a dynamic manufacturing environment. Customer requirements change rapidly and unpredictably in response to these technology advancements. A manufacturing enterprise must have the flexibility to adapt existing resources in a timely fashion to be competitive in this environment. The ability of a manufacturing enterprise to do this is currently limited by outmoded architectural paradigms and the lack of a fully integrated software toolset. The requirements that must be met by manufacturing enterprise architectures of the future are discussed. Gaps in the software toolset in support of manufacturing are observed in several contexts -- providing seamless interoperability across existing functional capabilities, providing a robust computational infrastructure to support manufacturing, and supporting a transition to the manufacturing enterprise architectures that will be needed in the future.

Key Words

Computer Aided Design (CAD); Computer Aided Engineering (CAE); Computer Aided Manufacturing (CAM); Computer Aided Process Planning (CAPP); Computer Integrated Manufacturing Open System Architecture (CIM-OSA); Computerized Numerical Control (CNC); Computational Infrastructure; Direct Numerical Control (DNC); Manufacturing Enterprise; Manufacturing Resources Planning (MRP II); Mass Customization; Material Requirements Planning (MRP); Object Oriented Analysis, Design, and Programming; Product Data Exchange using STEP (PDES); Standard for the Exchange of Product Model Data (STEP); Total Quality Management (TQM)

1. INTRODUCTION

In today's changing and highly competitive market, manufacturing enterprises can no longer afford to do business the way they did in the past. Rapidly advancing technology offers the promise of increased productivity and responsiveness; meanwhile, customer demands are dynamically changing at pace with this technology. Businesses that are willing to take advantage of this growing technology and dynamic market will survive; businesses that are not willing to change will be left behind.

The challenge is to effectively and efficiently apply the right technologies in the right combination for the right customer at the right time for the right strategic and tactical reasons in support of the right mission. Flexibility will be essential to accomplish this. Only in this way will a manufacturing enterprise be able to meet the sophisticated customer demands for quality products in a timely manner and at a competitive cost.

To meet this challenge, manufacturing enterprises will need to rethink and restructure the way they do business. They must evolve new enterprise architectural paradigms that integrate all elements of the enterprise (organization, manufacturing processes, information and computational infrastructure) in a holistic way. The architecture that a manufacturing enterprise adopts will form the basis for software toolsets to fill current gaps in software for manufacturing. Manufacturing enterprises will need to migrate from their current architecture to their future architecture in a carefully planned manner, in order to preserve "legacy" products, processes and information, and also to minimize operational disruption.

The process of changing the enterprise architecture must be one of planned revolution rather than evolution. Enterprises must develop the right vision, and use this as the basis for an organization that will realize this vision. Enterprises must be able to dynamically and flexibly bring their processes, capabilities and resources together to produce products on customer demand. To support this, enterprises must have an infrastructure of computer hardware, software, and information that will allow the enterprise to fully integrate existing software capabilities, to adapt these capabilities and resources to changing needs, and also to phase in new methods and supporting tools in an orderly fashion. In this way, new methods will be defined as they are needed and new tools acquired "just in time" to support these methods in response to customer requirements that are also defined and met "just in time".

Section 2 provides an overview of the state of the art in manufacturing enterprise architectures. Section 2.1 provides a historical perspective of the evolution of manufacturing enterprise architectures from several perspectives -- organization, culture, manufacturing process life cycle, and computational infrastructure. Gaps in the software that supports manufacturing are noted.

Section 2.2 discusses the needs of manufacturing enterprises of the future. The emphasis is on specifying requirements that future architectural paradigms of manufacturing enterprises must meet in order to be successful in a mass customization environment. A discussion of current activities that are under way to support the definition of future manufacturing enterprise architectural paradigms is included.

Finally, a summary of the gaps in software for manufacturing is provided in section 3. Concluding observations are provided in section 4.

2. Architecture State-of-the-Art

The state of the art in software for manufacturing can be characterized as a set of well-defined puzzle pieces that are waiting to be assembled and integrated into a coherent picture.

Section 2.1 discusses this current state of the art. Section 2.2 builds on this to present a view of how manufacturing enterprise architectures will need to evolve to meet the changing market environment of the future. This discussion will lay the foundation for the discussion of gaps in the state of the art that will be presented in section 3.

2.1 Manufacturing Enterprise Architectures Past and Present

In order to understand and characterize the state of the art in software for manufacturing as it applies to manufacturing enterprise architectures, it is useful to discuss manufacturing enterprise architectures from different perspectives.

The sections below discuss manufacturing enterprise architectures from three viewpoints -- the organizational structure (and underlying culture), the dynamic processes and flow through the manufacturing life cycle, and the underlying computational infrastructure.

2.1.1 Manufacturing Enterprise Culture and Organization

Manufacturing enterprises have followed a well-chronicled developmental evolution. The first significant advance from the historical age of individual trades and craftsmen was the industrial revolution, which was based on a "mass production" culture. The emphasis was on defining and implementing a streamlined linear flow of production to maximize efficiency, productivity and repeatability.

The manufacturing enterprise organizational structures reflected this style of production. Enterpreneurs invested in facilities that were revolutionary for their time, but were rigid and expensive. The risks involved with these investments were not shared with the production line workers and other staff, who as a result were not significantly involved in the key decisions of manufacturing enterprises.

Highly vertical, compartmentalized organizational structures reflected this inflexible management philosophy. Competition was encouraged, leading to organizational barriers and infighting. The result was empire building and a "that's not my job" response that inhibited problem resolution.

In recent times, two significant trends have led to the development of improved organizational structures. One was the rapid development and growth of a style of product line that was a departure from the heavy machinery products of the past -- namely the electronics industry. Rigid, capital-intensive manufacturing enterprise architectures were inappropriate for

this highly technical and rapidly changing industry. The flexibility to quickly and seamlessly reconfigure the tools and resources of production required a change from the rigid structures of the past.

The other development was the emergence of new ways of thinking about manufacturing enterprise organizations and culture. The term "Total Quality Management" (TQM) refers to a collection of principles defined by Deming [1], Juran [2] and others, including:

- clearly defined and communicated manufacturing enterprise mission and values that form the basis for strategic planning, marketing, research and development for the manufacturing enterprise;

- a clear focus on well-defined and communicated customer requirements, and incorporation of customer needs and feedback to adapt capabilities to the rapidly changing marketplace;

- team building, empowerment and partnership among employees and customers to resolve problems and improve processes ("do it right the first time", "fix the problem, not the blame");

- concurrent engineering to ensure design for manufacturing producibility; and

- continual improvement of products, processes and capabilities, based on quantitative measurement ("management by fact") supported by tools to provide modeling, simulation, process monitoring, statistical process control and related capabilities.

The visionary manufacturing enterprises in Japan and elsewhere effectively used these two developments to gain an advantage on their American counterparts. Manufacturing enterprises in the United States have engaged in intensive efforts to assimilate the philosophy and underlying culture of TQM, with mixed results. Meanwhile, the target is moving. As Pine, Victor and Boynton [3] point out, manufacturing enterprise cultures that are struggling to achieve TQM will need to take yet another evolutionary step to achieve the organizational culture, values and structures to support "mass customization". The implications of this on manufacturing enterprise organizations and architectures will be discussed in section 2.2.1.

2.1.2 Traditional Manufacturing Life Cycle

The organization and underlying culture for a manufacturing enterprise form the basis for the development of its manufacturing processes. The definition, implementation and improvement of these processes represent the most significant opportunities for the improvement and ultimate success of a manufacturing enterprise.

Mass production was traditionally based on the development of a set of linear, sequential steps required to manufacture a product. This traditional manufacturing enterprise process flow architecture is sometimes represented as a "waterfall" model, as shown in Figure 3.1.

In this model, the specification of product requirements is the first step. The requirements are then "thrown over the wall" to the design team to design the product. Once the product has been designed, including the assessment of producibility, logistics and other factors, a process plan is developed to specify the sequence of production steps and resources needed to manufacture the product. Schedules and dependencies are defined for the allocation of resources, including people, machinery, shop floor and inventory of parts (which frequently are stocked excessively in anticipation of needs before they are really defined). Then the product goes through the sequence of steps (drill, mill, stamp, heat treat, deburr, polish, etc.) on the job floor that are necessary to manufacture it. The product is inspected, tested, packaged and delivered.

Support services include marketing, capacity planning, purchasing, receiving and inventory. These support services are frequently provided by parts of the manufacturing enterprise that are separate from the production work flow path. Quality control is typically injected, as if by magic, at various "check points" in the manufacturing life cycle, by people who are not directly involved in the manufacturing processes.

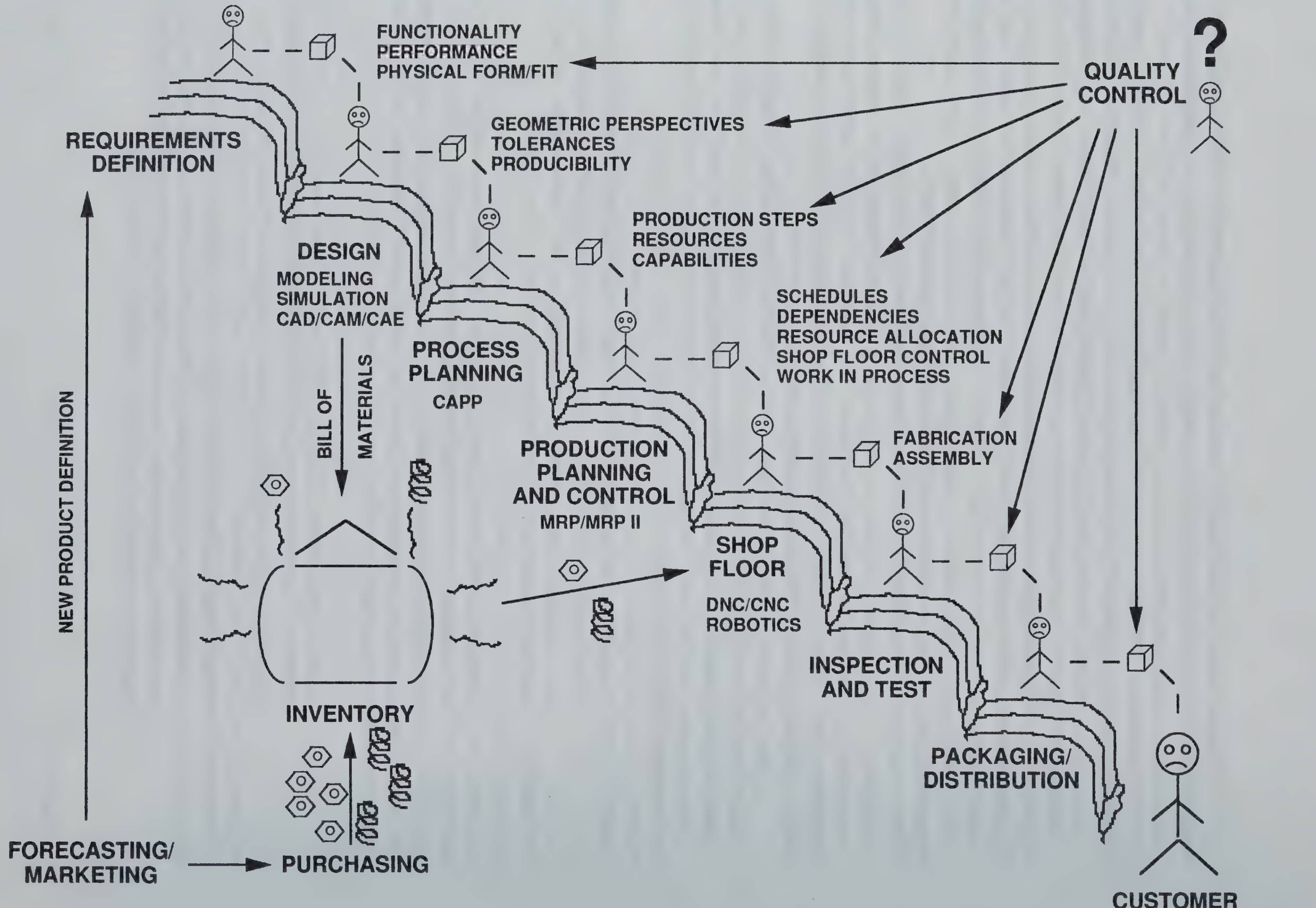
Most manufacturing enterprises have observed that the capabilities provided by software toolsets offer immense promise to enhance the quality, cost effectiveness and productivity of the manufacturing process. The list of software capabilities listed below is by no means comprehensive, but indicative of the style of the software capability set that is available to manufacturing enterprises today.

- Computer Aided Design (CAD), Computer Aided Engineering (CAE), Computer Aided Manufacturing (CAM) -- This software tool family provides capabilities that aid designers in the generation, specification and representation of designs, including geometric perspectives, tolerances, etc.

- Computer Aided Process Planning (CAPP) -- This software tool capability set uses the output of CAD/CAM/CAE software tools to generate a process plan to manufacture the designed product. The process plan is generated from a sequence of atomic process steps in a manner analogous to the creation of a computer program from a set of instructions.

- Material Requirements Planning (MRP) and Manufacturing Resources Planning (MRP II) -- This software toolset provides capabilities such as routing, scheduling, resource allocation and shop floor control in support of the manufacturing process.

- Direct Numerical Control (DNC), Computerized Numerical Control (CNC), and related machine tool programming -- This software toolset executes directly on the manufacturing machines and commands the machines to perform the processing operations according to the machine capabilities and tolerances.



FUNCTIONALITY
PERFORMANCE
PHYSICAL FORM/FIT
QUALITY CONTROL
?
REQUIREMENTS DEFINITION
GEOMETRIC PERSPECTIVES
TOLERANCES
PRODUCIBILITY
PRODUCTION STEPS
RESOURCES
CAPABILITIES
DESIGN
MODELING
SIMULATION
CAD/CAM/CAE
SCHEDULES
DEPENDENCIES
RESOURCE ALLOCATION
SHOP FLOOR CONTROL
WORK IN PROCESS
PROCESS PLANNING
CAPP
BILL OF MATERIALS
NEW PRODUCT DEFINITION
PRODUCTION PLANNING AND CONTROL
MRP/MRP II
FABRICATION
ASSEMBLY
SHOP FLOOR
DNC/CNC
ROBOTICS
INSPECTION AND TEST
INVENTORY
PACKAGING/ DISTRIBUTION
FORECASTING/ MARKETING
PURCHASING
CUSTOMER

These and other software toolsets have been combined (but not integrated!) as part of a strategy to implement a "paperless factory". Indeed, this term has been used to characterize a widely-varying degree of manufacturing automation and software support. The fact is that true integration of the elemental steps of this traditional manufacturing cycle has not been achieved.

In fact, the software for manufacturing state of the art can be characterized by the following observations:

- o numerous software capabilities exist, and several are well developed as isolated software tools;
- o some software toolsets (e.g., MRP II, DNC) are either not widely available on a robust variety of hardware/software platforms or are not suitable for use across the wide range of enterprises from the very large to the very small;
- o the transition from one toolset to the next in the sequential manufacturing process usually involves manual steps (i.e., the tools are typically not interoperable or integrated); and
- o the sequential view of manufacturing is itself a gap, with far-reaching implications that will be discussed further in section 2.2.2.

2.1.3 Software Infrastructure Evolution

To take full advantage of the promise offered by software to improve all aspects of manufacturing, a computational infrastructure is needed that will serve as the platform upon which the software tools will run. This infrastructure must have the following attributes:

- o computing hardware and system software with sufficient processing and storage capacity for the enterprise manufacturing software needs;
- o a user interface that is easy to learn and use and naturally reflects the style of thought that is characteristic of the enterprise;
- o communications capabilities that permit the exchange of information between application entities (e.g., clients and servers) and between sites;
- o the provision of interoperability across heterogeneous platform computing elements; and
- o capabilities that support extensibility and growth so that "legacy" software and data products are not lost.

These general capabilities are already available. Despite this, many manufacturing enterprises have not incorporated them into their business. Thus, this level of computational infrastructure does not represent a gap in the state of the art, but rather a gap in the culture, or cost, or technology, depending on the enterprise.

There are, however, software gaps in the manufacturing enterprise computational infrastructure state of the art. These gaps pertain to the development of intelligent services that promote software toolset interoperability. The requirements for these services are discussed in section 2.2.3, and the gaps are discussed in section 3.

2.2 Future Directions for Manufacturing Enterprise Architectures

With the discussion of the state of the art in section 2.1 serving as a background, the directions that must be pursued by successful manufacturing enterprises of the future are discussed in the sections that follow.

Each of the three perspectives of manufacturing enterprise architecture (organization, process flow, infrastructure) will be revisited and discussed in the context of the requirements that must be met to be successful in the business environment of the future. Some of the activities that have been initiated by manufacturing standards organizations are also discussed.

2.2.1 Mass Customization Cultural Revolution

As noted in section 2.1.1, the evolution of manufacturing enterprise organizational structures from handicraft through mass production to TQM now needs to take another step. Pine, Victor and Boynton [3] discuss the limitations of TQM and process improvement in a mass customization environment. They observed that organizations based on process improvement are frequently not flexible enough for mass customization. Process improvement teams tend to take on a life of their own, creating many of the same barriers that TQM originally set out to avoid.

Manufacturing enterprises in a mass customization environment need to start with their vision and values and build organizational philosophies and structures that enable flexibility. Pine, Victor and Boynton suggest that values based on "being the best" should be replaced with values based on building "what the customer wants, when and where the customer wants it". Focusing on the optimization of processes that may quickly become outmoded as customer requirements move in different, unanticipated directions is wasteful. Market research will still be an important guide to the direction of the enterprise, but it will become increasingly important to adjust in real time to unanticipated changes in technology and customer requirements.

With values that focus on mass customization as the basis, the organizational structures of manufacturing enterprises will tend to treat all of the company resources as multi-dimensional and unconstrained by artificial department boundaries or matrix project structures. Personal capability sets will replace job descriptions, and problem solving teams will be instantaneously created based on these capabilities and disbanded to be reconfigured for the next project when their work is done.

Loosely coupled structures will be important at all levels. If the enterprise starts with values of flexibility and adaptability, this mindset will permeate the organization, the process flow (see section 2.2.2) and the computational infrastructure (see section 2.2.3). Indeed, good organizations, good manufacturing process flows, and good software architectures have in common a loosely coupled, flexible and easily reconfigurable structure.

In the words of Pine, Victor and Boynton, the organization of an effective manufacturing enterprise in a mass customization environment will be:

- instantaneous (enabling enterprise processes and capabilities to be reconfigured quickly);
- costless (to configure the instant teams, with some investment costs for establishing and evolving the computational infrastructure);
- seamless (enabling resource reconfiguration to be smooth); and
- frictionless (providing enterprise organizations and supporting software tools to enable teams of people who have never met to work together effectively).

2.2.2 Manufacturing Life Cycle Future Requirements

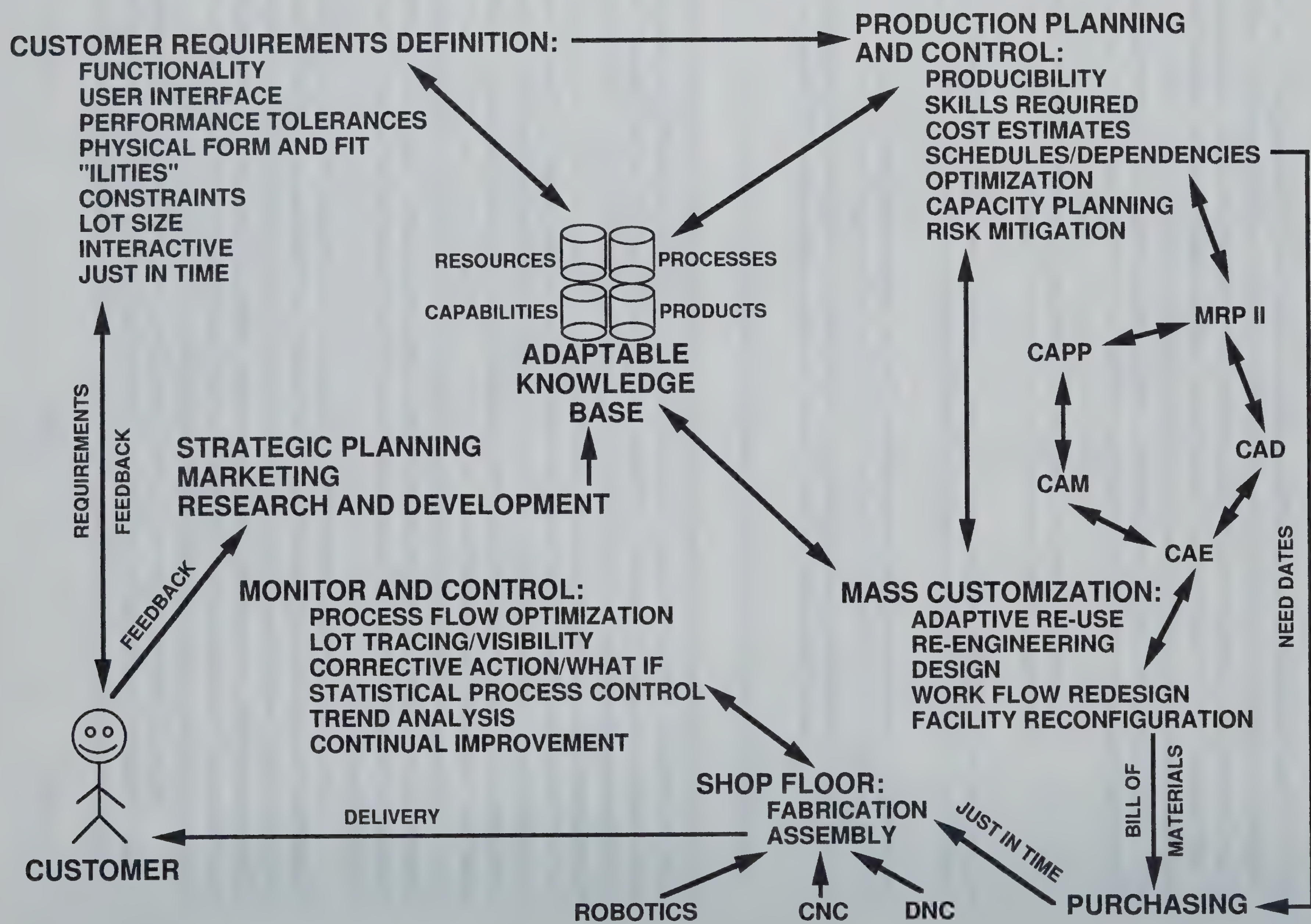
The requirements for the manufacturing enterprise architectures of the future will be driven by the flexibility needed to reuse, adapt and reconfigure existing manufacturing enterprise processes, resources, and products to manufacture new products in response to often unpredictable customer requirements (mass customization). As noted above, this must be done in an instantaneous, costless, seamless, and frictionless manner.

The ultimate goal would be to establish a "lot 1" culture within the manufacturing enterprise; i.e., to enable the enterprise to use its capability set to define the processes, configure the resources and produce an arbitrarily small number of customized products quickly, at low cost and with high quality and reliability.

The organizational and operational paradigms of manufacturing enterprises of today are outmoded in their ability to meet the requirements of mass customization. The "waterfall" model must be replaced by a fully integrated manufacturing life cycle that institutionalizes concurrent engineering.

Each manufacturing enterprise will need a well-planned migration strategy that will first establish linkages to integrate existing manufacturing software tools, provide translation capabilities to preserve key "legacy" software tools and information, and finally provide an architecture of fully automated and interoperable manufacturing software tools. "Interoperability" is the ability of software tools to communicate with, and operate synergistically with, other software tools both internal and external to the enterprise.

Figure 3.2 presents a view of a manufacturing life cycle of the future. Such a life cycle must be based on identification of customer requirements "just in time". This requirements focus is absolutely essential. As Pine, Victor and Boynton [3] point out, enterprises that try to project


CUSTOMER REQUIREMENTS DEFINITION:
FUNCTIONALITY
USER INTERFACE
PERFORMANCE TOLERANCES
PHYSICAL FORM AND FIT
"ILITIES"
CONSTRAINTS
LOT SIZE
INTERACTIVE
JUST IN TIME
PRODUCTION PLANNING AND CONTROL:
PRODUCIBILITY
SKILLS REQUIRED
COST ESTIMATES
SCHEDULES/DEPENDENCIES
OPTIMIZATION
CAPACITY PLANNING
RISK MITIGATION
RESOURCES
PROCESSES
CAPABILITIES
PRODUCTS
ADAPTABLE KNOWLEDGE BASE
MRP II
CAPP
CAD
CAM
CAE
REQUIREMENTS
FEEDBACK
STRATEGIC PLANNING
MARKETING
RESEARCH AND DEVELOPMENT
FEEDBACK
MONITOR AND CONTROL:
PROCESS FLOW OPTIMIZATION
LOT TRACING/VISIBILITY
CORRECTIVE ACTION/WHAT IF
STATISTICAL PROCESS CONTROL
TREND ANALYSIS
CONTINUAL IMPROVEMENT
MASS CUSTOMIZATION:
ADAPTIVE RE-USE
RE-ENGINEERING
DESIGN
WORK FLOW REDESIGN
FACILITY RECONFIGURATION
NEED DATES
SHOP FLOOR:
FABRICATION
ASSEMBLY
DELIVERY
CUSTOMER
JUST IN TIME
BILL OF MATERIALS
ROBOTICS
CNC
DNC
PURCHASING

requirements too far in the future have been less successful than enterprises that focus on the flexibility to meet requirements that are defined "just in time".

Gause and Weinberg [4] provide valuable insights into the art of defining good requirements. Requirements definition is multi-dimensional. For example, requirements may involve customer interface, functionality, performance, safety, reliability, and/or physical form and fit. Requirements may be completely new, or they may be modifications of requirements that have been previously defined in a different context.

A flexible and user-friendly two-way link will be needed that will guide the customer through a definition of requirements. Such a software tool could then feed back a view (graphical demonstration and/or textual information) of proposed functional and physical capabilities of a customized product in response to the information entered by the customer. This will allow the customer to interact with product definition experts within the manufacturing enterprise to quickly iterate and converge on a customized product that is achievable within the cost, schedule and quality requirements of the customer. In effect, this would be an automated prototyping and marketing demonstration tool that interactively matches enterprise capabilities with requirements that are explicitly tailored to each individual customer.

The manufacturing enterprise software toolset would then concurrently and iteratively perform the following capabilities:

- assess the capabilities of the manufacturing enterprise to meet the defined requirements;
- determine the producibility of the specified product;
- estimate and optimize the cost and schedule according to enterprise and customer specified constraints;
- assess and update the impact of this manufacturing process on capacity planning;
- support risk analysis and definition of risk mitigation strategies and alternatives;
- define the manufacturing process, tailoring and reconfiguring existing enterprise manufacturing resources, including the factory floor configuration; and
- define and schedule the interdependent manufacturing process steps, including truly "just in time" inventory and resource allocation.

As the product moves through the manufacturing cycle, software toolsets will provide full visibility and traceability of the manufactured lot and the manufacturing processes used. Software toolsets providing CAD, CAPP, MRP II and shop floor capabilities such as numerical control will be fully integrated and interoperable.

Monitor and control capabilities will include automated initiation of software toolsets "just in time". Statistical process control will be accomplished by collecting productivity and quality

metrics that can be used for real-time analysis of errors and bottlenecks as well as for off-line benchmarking and trend analysis to support process improvement. Intelligence built into the products themselves can be used to program in new features, capture fault data, and interoperate with other software tools to support corrective action in real time. For example, monitors could provide an interactive "what if" capability for shop floor supervisory personnel to identify and initiate contingency actions.

Testing should not be considered an after-the-fact "injection" of quality into the product, but should be an ongoing process to verify that the customer requirements are satisfied. Intelligence embedded within products will be very promising to provide this capability. Timely and cost-effective distribution and delivery systems that eliminate the "middle man" will also be essential.

Finally, it will be essential to provide automated customer support capabilities and obtain customer feedback regarding satisfaction with the cost, schedule and quality of the delivered lot size. Software embedded within the delivered product will be able to provide powerful capabilities to analyze the feature usage patterns as well as the reliability of the product. This information can then be transmitted electronically back to the manufacturing enterprise, helping it to focus on ways to improve the products as well as to provide possible future product developments and feature enhancements. This information in turn could be fed back into the strategic planning of the enterprise to focus research and development investments most effectively on the technologies that will enhance the mass customization capabilities needed in the future.

2.2.3 Software Infrastructure Future Requirements

The need for an open, flexible architecture for manufacturing enterprises has been clearly identified. Such an architecture must provide a computational infrastructure that will:

- enable seamless interoperability of different software applications at the same site, at different sites of the same "virtual' manufacturing enterprise, and at different manufacturing enterprises;
- enable flexibility of scale;
- enable technology insertion for extensibility and growth;
- provide translation and migration support between current and future configurations in a way that preserves key "legacy" software tools and data;
- feature a powerful, user-friendly interface into the network capability set and a comprehensive view of the software tools available to support manufacturing; and
- enable the adaptive reuse of existing manufacturing enterprise software tools, information, capabilities, processes and products in support of mass customization.

A hierarchical layering of open network and application architectural paradigms, as shown in Figure 3.3, is suggested to meet these requirements. Each layer provides an increasingly application-specific and user-oriented functional capability set. Within each layer (especially within the upper layer of application software tools), the structure is "flat", with each software tool providing well-defined and standard interfaces that enforce access and visibility rules. This is an "object oriented" paradigm that promotes reliability and flexibility through information hiding. Consult Booch [5] for an in-depth discussion of object oriented analysis, design and programming.

The computer operating systems form the most basic layer of the architecture. In practice, a set of operating systems are layered on top of each other to provide increasing functionality and user interface capabilities.

The next layer above the operating systems provides communication services. This layer is also hierarchical, with typically homogeneous systems communicating within work areas; a local area network, perhaps with gateways or bridges, connecting potentially heteogeneous work areas at a site within an enterprise, and a wide area network connecting to geographically distant sites and/or other enterprises.

The next layer above communication services provides system and communications management, including user security. With the exception of some aspects of system management, the technology in these lower layers is well understood and will not be discussed further.

The two uppermost layers, however, are not well defined. The layer above system management is called the "enabling layer". Its purpose is to provide the following functionality at an application layer:

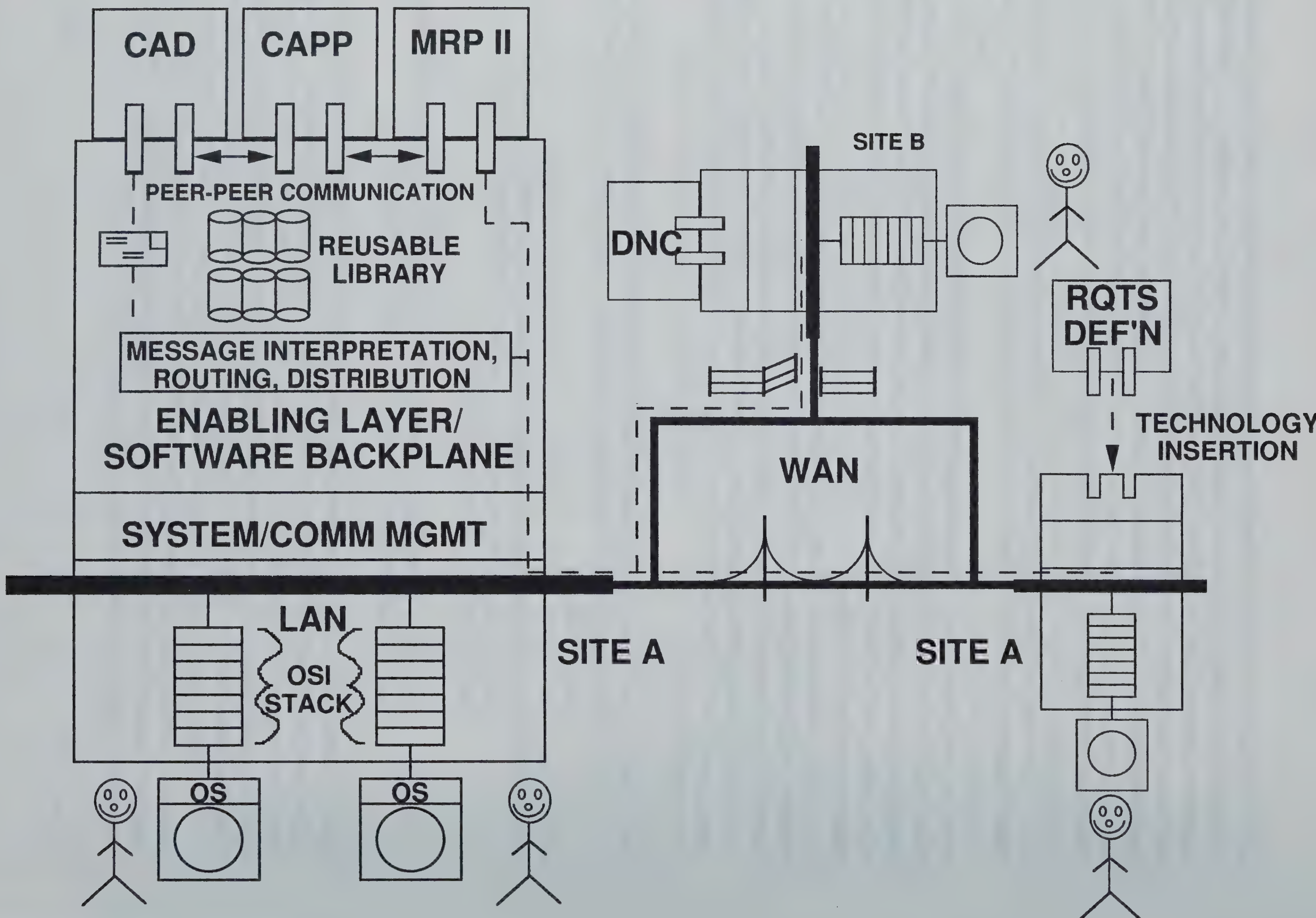
- o intelligent message interpretation and distribution to appropriate applications, including generic mail capabilities as well as support for application-specific communication;

- o intelligent database services that not only provide key enterprise system information within a (virtual) repository, but also provide a common semantic interpretation of the information within the repository; and

- o notification to subscriber applications of key enterprise system events and changes in the values of data.

The definition and implementation of this "manufacturing enterprise enabling layer" is a significant gap in the current state of the art in automated manufacturing enterprises. This layer may also be regarded as an intelligent "software backplane" that provides the "glue" between the generic lower layers and the enterprise-specific upper layers. (Note that this implies that this "enabling layer" needs to embody both generic and enterprise-specific knowledge and features).

Finally, the uppermost layer contains the specific applications. Note that each application is shown with a "plug in" module. The module "plugs in" to the enabling layer via its well-defined interfaces and visibility rules. Communication may be peer-to-peer if well-defined and/or application-specific, or it may use a message paradigm and go through the enabling layer if it is


CAD
CAPP
MRP II
PEER-PEER COMMUNICATION
REUSABLE LIBRARY
MESSAGE INTERPRETATION, ROUTING, DISTRIBUTION
ENABLING LAYER/ SOFTWARE BACKPLANE
SYSTEM/COMM MGMT
LAN
OSI STACK
OS
OS
SITE A
SITE B
DNC
RQTS DEF'N
TECHNOLOGY INSERTION
WAN
SITE A

generic and may potentially change as the manufacturing enterprise matures. Such a generic message paradigm promotes flexibility, adaptability and growth via loose coupling of the enterprise software tools.

An information repository is essential to complete this manufacturing enterprise architecture. This repository (which may be distributed) would contain the key enterprise data (capabilities, resources, processes, products, services, customer information, product reliability data, process improvemetn data, etc.). The data would reflect standardized semantics (see section 2.2.4 for relevant standards definition activities). The data would also be parameterized to reflect (for example) product functional capabilities, product physical envelope, and interrelationships among data entities (e.g., processes and capabilities required to manufacture a product).

Note also that software and data should ultimately be embedded within products in the field to provide the robust capabilities discussed in section 2.2.2.

2.2.4 Current and Future Directions

A number of activities have been initiated by various manufacturing standards organizations to define architectural and information structure frameworks. These frameworks will form the foundations for the manufacturing enterprise architectures of the future. This will fill significant conceptual gaps and allow for the insertion of software toolsets to fill gaps in the automation of manufacturing processes.

Before discussing the standards definition activities for software architectures for manufacturing, it is useful to note the work of Zachman [6], and later Sowa and Zachman [7]. They define a two-dimensional set of architectures for software application systems. One dimension represents increasing levels of detail (from user/planner to owner to designer to builder to subcontractor/implementer); the other dimension covers what (objects), how (functions/processes), where (site locations/nodes), when (schedules/processing dependencies), who (user/developer organizations), and why (vision and mission). This model applies to any software system, not just to manufacturing enterprise environments.

The Computer Integrated Manufacturing Open System Architecture (CIM-OSA) [11] defines a three-dimensional approach to software architectures for manufacturing. One dimension involves stepwise generation from a functional view (how) to an information view (what) to a resource view (where) to an organizational view (who). A second dimension involves stepwise derivation from requirements definition to design specification to implementation description, analogous to Zachman's increasing levels of detail. The third dimension of CIM-OSA is stepwise instantiation from a generic product line to a partial product family (e.g., a set of products that differ only in some parameters), and finally to a particular specific product. The added value of this third dimension of-CIM-OSA is the focus on commonality of products and processes within a manufacturing enterprise and the capability to quickly recombine existing processes and resources to produce new products on customer demand.

Other sources of information regarding future directions of software architectures for manufacturing include the "New Manufacturing Enterprise Wheel" from the Computer and Automation Systems Association (CASA) of the Society of Manufacturing Engineers (SME) [9]. Also of interest are Halevi [10], Scheer [11], and Thacker [12].

In addition to the need to define standard manufacturing enterprise architectural frameworks, it is also essential to define standard terminology, semantics and access methods for manufacturing enterprise information. The current state of the art in this area is such that the same or similar entities have different names across manufacturing enterprises and supporting software applications. Worse, there is the potential that a common name may have a different meaning in different manufacturing contexts.

Activities are under way to define common information semantics and access methods across manufacturing enterprises and software application toolsets, notably the Standard for the Exchange of Product Model Data (STEP) [13] and Product Data Exchange using STEP (PDES) [13].

Once defined, enterprise information standardization can serve as a vehicle for mass customization. Ultimately, each manufacturing entity will have standard semantics augmented by parametric information to allow for customization. This information, including parametric variations, can then be "catalogued" in an information repository, allowing for parameterized access, use, modification and storage of the modifications required for a customized product.

3. Architecture Gaps

The discussion in section 2 pointed out that there are gaps in the current state of the art in software to support manufacturing enterprise architectures. Some of these gaps are functional limitations, including some that result from operational manufacturing paradigms that are becoming outmoded. Some gaps are cross-functional; i.e., they pertain to the unrealized potential for automating interaction across existing functional software tools. Some gaps refer to platform and scaling limitations.

The gaps in software to support manufacturing enterprise architectures and functional capabilities are discussed below.

Interactive customer requirements definition

As discussed in section 2, the mass customization manufacturing enterprises of the future will need the means to actively solicit emerging customer requirements in a timely fashion. It must be possible to interact with a customer (who may not be a sophisticated computer user) and help that customer specify the functional and physical characteristics that the customer needs. This interaction should help the customer define and quantify what is really needed, and should relate these requirements to the products and capabilities that the manufacturing enterprise possesses.

Although the details of the toolsets that will support this functionality will vary with the business scope or "niche" of the manufacturing enterprise, there is great potential for providing a software toolset with common capabilities across manufacturing enterprises that isolates the "niche" data in databases.

A software toolset is needed that will provide the following services in support of interactive definition of customer requirements:

- o interactive dialogue with a customer to guide the definition of quantified requirements, including functional capabilities and physical requirements;
- o interactive dialogue with the customer to match his requirements with existing products and capabilities of the manufacturing enterprise;
- o definition and capture of customer requirements in a form that is usable to assess producibility and to support forecasting, capacity planning, computer aided design, process planning and other steps in the manufacturing cycle;
- o assessment of the cost, schedule and performance implications of the customer-required product;
- o interactive graphics and prototyping tools that show the customer possible implementations of his requirements; and
- o capture of product improvement feedback from customers.

Automated linkage between customer requirements, forecasting and capacity planning

As discussed in section 2, a few companies have had some success implementing a mass customization manufacturing cycle. There is an opportunity to use the output from quick interactive customer requirements definition tools as input to forecasting and capacity planning tools.

A software toolset is needed that will provide the following services in support of automating the linkage between the output from interactive customer requirements definition support tools and forecasting and capacity planning applications:

- o capture of new and modified product information in the enterprise data repositories;
- o trend analysis and prototyping tools to support forecasting of the market for the same or similar product variations; and
- o use of trend analysis information to direct marketing and prototyping efforts that focus on the enterprise capabilities to produce the product that the customer wants when he wants it, consistent with the mission and strategic objectives of the manufacturing enterprise and with profitability.

Automated linkage between customer requirements, forecasting, purchasing and inventory

A key to the successful implementation of mass customization and improved efficiency will be the implementation of purchasing and inventory policies that provide the required parts, materials and other resources (including human resources) "just in time". As discussed in section 2, manufacturing enterprises have yet to incorporate a true "just in time" philosophy into their corporate policies and supporting software systems.

A software toolset is needed that will provide the following services in support of automating the linkage between the output from interactive customer requirements definition support, forecasting and scheduling tools with purchasing and inventory applications:

- o adjustment of tentative acquisition schedules according to the strategic and forecasting implications of new customer requirements; and

- o definition of "hard" need dates for parts and materials to support schedules that are truly "just in time" to meet the customer's needs.

Automated match of requirements to capabilities

The incorporation of capability data sets covering all of the resources of the manufacturing enterprise must be done in such a way that the applicability (and hence the effectively customized reuse) of enterprise capabilities can be assessed in an automated fashion as part of the production planning cycle for the new customer product.

A software toolset is needed that will provide the following services in support of automating the linkage between the output from interactive customer requirements definition support tools and capability and risk assessment applications:

- o assessment of the producibility of the new product within the customer's performance, schedule and cost constraints;

- o assessment of key performance, schedule and cost risks associated with manufacturing the new product; and

- o generation of performance, cost and schedule estimates within the constraints specified by the customer.

Automated linkage between CAD and CAPP

As discussed in section 2, computer aided design (CAD) software toolsets have existed and have been used for many years. Also as noted in section 2, computer aided process planning (CAPP) toolsets have been available for many years, providing the capabilities to build process plans from computer aided design information. Section 2 points out that there is a software tool gap between the generation of CAD output and the initiation of CAPP software applications.

A software toolset is needed that will provide the following services in support of automating the linkage between CAD design output and CAPP software applications:

- o use the engineering design outputs of CAD software applications, together with process requirements for the designed product and fabrication and assembly machine capability data, to generate process plans that are "optimal" according to selected optimization criteria (e.g., schedule, machine availability, machine utilization time, cost effective use of machines based on depreciation and other factors, total manufacturing throughput cost of the production cycle);

- o tie this link into other manufacturing enterprise software and database capabilities (e.g., use database attribute information to promote reuse of previously generated process plans, use information from monitors to adjust generated process plans to improve the manufacturing cycle); and

- o automatically schedule and initiate execution of CAPP software "just in time" to support the generation of optimized process plans.

<u>Automated linkage between CAPP and MRP II</u>

As discussed in section 2, CAPP and MRP II software toolsets have been available for a number of years. Section 2 points out that there is a software tool gap between the generation of CAPP output and the initiation of MRP II software applications.

A software toolset is needed that will provide the following services in support of automating the linkage between CAPP output and MRP II software applications:

- o use the process plan outputs of CAPP applications, together with scheduling constraint data, to generate part and material routing and flow information, schedule use of fabrication and assembly machines and processes, generate contingency schedules, etc.;

- o tie this link into other manufacturing enterprise software and database capabilities (e.g., create "just in time" schedules for enterprise materials and resources, use information from monitors to adjust schedules and machine usage to improve the manufacturing cycle); and

- o automatically schedule and initiate execution of MRP II software "just in time" to support the generation of optimized schedules and resource utilization.

Automated linkage between MRP II and the shop floor

As discussed in section 2, MRP II and DNC software toolsets have been available for a number of years. Section 2 points out that there is a software tool gap between the generation of MRP II output and the initiation of DNC and other machine software applications.

A software toolset is needed that will provide the following services in support of automating the linkage between MRP II output and DNC software applications:

- use the schedules, routing and other outputs of MRP II applications, together with CAD and CAPP outputs, to generate the information needed by DNC and other automated fabrication, assembly and shop floor control software;

- tie this link into other manufacturing enterprise software and database capabilities (e.g., use information from monitors to improve the manufacturing cycle); and

- automatically schedule and initiate execution of DNC software "just in time" to perform the appropriate manufacturing fabrication and assembly steps.

Automated linkage between the shop floor and monitoring tools

Although some fabrication and assembly monitoring capabilities currently exist, these capabilities need to be more fully integrated with other manufacturing software application toolsets. Such monitoring software tools, when fully integrated into the manufacturing cycle, will provide a full suite of data capture, analysis and feedback capabilities. These capabilities will not only pinpoint problems with the current manufacturing lot, but will also provide the basis for process improvement.

A software toolset is needed that will provide the following services in support of integration of monitoring capabilities with the software toolset that supports requirements specification, design, process planning, fabrication and assembly functions:

- tracing of products through the manufacturing cycle to monitor both the product quality and the performance of the manufacturing cycle;

- identification and automated capture of key performance metrics (e.g., timing, tolerance and error data) from DNC machines and other fabrication and assembly machines that have resident software application toolsets;

- analysis of data to flag error or out-of-tolerance conditions, determine process bottlenecks, and perform statistical process control (SPC) functions;

- identification of possible contingency plans or process adjustments in response to problems observed by monitoring software tools;

- o use of prototyping, modeling and/or metric tools to generate "what if" scenarios that assess implications (technical, schedule, cost, risk) of manufacturing process modifications before they are implemented; and

- o feedback of analysis data for use by software tools in all parts of the manufacturing cycle, including redefinition of CAPP and/or MRP II planning, scheduling and resource allocation outputs.

Flexible manufacturing centers

As discussed in section 2, a mass customization culture implies the ability to allocate, reuse and reconfigure resources efficiently to support the manufacture of new or modified products.

A software toolset is needed that will provide the following services in support of flexibility of a manufacturing enterprise:

- o definition of resource modifications to support new product manufacturing requirements;

- o prototyping capability set to optimize reconfigurations and other resource allocations according to specified criteria (e.g., schedule, machine availability, machine utilization time, cost effective use of machines based on depreciation and other factors, total manufacturing throughput cost of the production cycle); and

- o reconfiguration of the facility (factory floor, assembly line, production cell, etc.) as appropriate to manufacture the new product in an optimal manner.

Enterprise domain definition and standardization

To fill many of the gaps noted above (those that pertain to interoperability across manufacturing software toolsets), the ability of the software toolsets to communicate with each other in a commonly understood manner will be necessary. As noted in section 2, the current state of the art is such that some data items have nonstandard names; also, in the absence of standards, there is the risk that a data name may have a slightly different meaning (or at least a different format) in different contexts.

As noted in section 2, various manufacturing standards definition activities are under way. As these standards become defined by the cognizant manufacturing societies, software tools can be created and/or tailored to use these standards to populate databases in a consistent and commonly understood manner.

A software toolset is needed that will provide the following services in support of consistent naming and use of data items contained within the manufacturing enterprise data repositories:

- o standardization of data naming and classification;

- specification of key semantic attributes (e.g., processes or capabilities required by products, product performance constraints);

- parameterization of the semantic attributes to allow for full utilization of process tailoring capabilities in response to customer definition of new product variations; and

- interactive user definition support to specify the required values of parameters.

<u>Distributed, heterogeneous "virtual enterprise" capability set</u>

As noted in section 2, even some of the most advanced manufacturing enterprises have physical "islands of automation" that are not interfaced, even though the capability to do so is readily available. The goal, which is demonstrably achievable in the near term, is to establish an electronic "virtual enterprise" that allows all of the elements of the manufacturing enterprise to be physically located where it makes the most sense from a business perspective. (E.g., allow marketing to be located near the customer while manufacturing is located near the physical resources, perhaps in a different part of the world.)

A software toolset is needed that will provide the following services in support of implementing a true "virtual enterprise"; i.e., providing an electronic interface capability set that does not penalize the enterprise in any way for distributing its functional capabilities according to the rules of common business sense:

- interoperability of elements of the same software toolset (such as clients with a server), or of logically interrelated software toolsets, that are implemented on heterogeneous platforms; and

- integration of software toolsets that are distributed across geographically discontiguous sites.

<u>Intelligent data services</u>

For the reasons given above in the discussion of enterprise domain definition and standardization, the ability to resolve both the format and the interpretation of data is needed across manufacturing enterprise software toolsets. Until the definition and implementation of common names, including specification of semantic attributes as well as formats, has been completely standardized, the ability to resolve existing format and semantic differences in data that is passed between manufacturing software application toolsets will need to be provided.

A software toolset is needed that will provide the following services in support of commonly understood and unambiguous communication between existing or future manufacturing software applications ("intelligent data services"):

- resolution of difference in data format and semantics between software applications (e.g., data translation);

- data and command routing services to move information and control among software application toolsets in an automated fashion; and

- migration and data translation services that will convert data and command formats between existing "legacy" applications and the more fully integrated application toolsets of the future.

Intelligent products

As discussed in section 2, the potential for mass customization will not be fully realized without incorporating software into the product line. Software embedded within products will enhance the capabilities for the products to synergistically interact with other software tools, so that the products become an integral part of the manufacturing enterprise architecture. Software tools embedded within products can provide monitoring, diagnostic and logistic support capabilities, and even provide a form of "micro-programmability" that will provide virtually instantaneous customization of existing products to meet newly defined customer needs. This micro-programmability can also be used to facilitate prototyping that can assist strategic planning and forecasting.

A software toolset is needed that will provide the following services in support of intelligent product capabilities:

- product interaction with the manufacturing cycle to provide monitoring and verification capabilities;

- built-in test capabilities to pinpoint faults, record information that can be used for product and process improvement, and time-driven indicators to initiate self-test, preventive maintenance and other logistic functionality;

- fault tolerant features including capabilities to avoid faults, correct faults, and/or to perform essential mission and safety functional capabilities in the presence of faults;

- on-line "news" capabilities (e.g., electronic mail) that notify product owners of new product developments, problems, etc.; and

- reprogrammability features that allow an existing product to be readily extensible to provide new, perhaps unanticipated functionality in response to new customer needs as they are defined.

Seamless incorporation of new technologies and capabilities

As discussed in section 2, an essential attribute of any successful manufacturing enterprise of the future will be the flexibility to reconfigure in the light of technological advances. This attribute will stand in sharp contrast to the rigid structures of traditional manufacturing enterprises that inflexibly build the "old way of doing things" into all aspects of the manufacturing architecture.

A capability set is needed within manufacturing enterprise architectures of the future to readily incorporate technological advances in software technology, manufacturing technology and product line technology.

The structural implications of this consideration on manufacturing enterprise architectures are discussed in section 2. In addition, a software toolset is needed that will provide the following services in support of the capability for a manufacturing enterprise to "seamlessly" incorporate technological advances:

- use of enterprise domain data definition capabilities and the extensibility attributes of a robust set of manufacturing software tools to incorporate a new capability set;

- use of flexible manufacturing center capabilities to reconfigure and reallocate resources to incorporate new technologies; and

- use of embedded product software capabilities to reprogram products to incorporate new technologies.

Fully integrated enterprise manufacturing cycle

As discussed in section 2, many of the gaps in the current state of the art in software for manufacturing result from a traditional "waterfall" view of the manufacturing cycle. This view is a linear progression of steps in the cycle, with gaps resulting in many cases from the lack of automated continuity between successive steps in the cycle.

Section 2 points out that this architecture itself represents a significant gap in the state of the art. The mass customization manufacturing enterprises of the future will require a new paradigm that will feature more synergistic interaction among all the manufacturing "phases". These phases will interact in a continual, nonlinear fashion to achieve concurrent engineering and mass customization. The resulting manufacturing enterprise architecture and supporting software toolset will fully integrate, rather than merely interface, existing and future software tools.

In addition to merging the "horizontal" architectural elements, the manufacturing enterprise architectures of the future will integrate the "vertical" views of the architecture (enterprise architecture, manufacturing processes, product and capability domains, infrastructure).

A software toolset is needed that will provide the following services in support of fully integrating the manufacturing enterprise architectures of the future in a concurrent engineering fashion:

- using customer requirements information to drive prototyping, marketing, research and development, capacity planning, purchasing and production activities;

- automating design and process planning activities, including reuse of existing capabilities and real-time modification of intelligent product capabilities, based on definition of customer requirements "just in time";

- optimizing manufacturing configuration, production, machine utilization, activity based cost, and human resource utilization based on requirements, design, customer constraints and enterprise resource limitations;

- automated initiation, monitoring and control of the production process flow;

- feedback of error and bottleneck information for corrective action, process improvement and research and development direction; and

- seamless incorporation of new product definitions, research and development innovations, monitored information (including information received from intelligent products in the field) and customer feedback into the manufacturing enterprise information repositories.

A truly concurrent architecture, with software tools providing some or all of the capabilities above, may be viewed as analogous to a computer aided process planning software toolset. CAPP tools treat process steps as modules that are used to generate a manufacturing process for a newly designed product. The capability set listed above would (among other things) treat all of the manufacturing resources of the enterprise as modules that are combined to produce a newly customized product in an instantaneous, costless, seamless and frictionless manner on customer demand.

<u>NOTE:</u> There are also numerous related non-software gaps that must be filled, involving enterprise organizational and operational paradigms, adoption of mass customization culture (including a true "just in time" policy set), and effective use of capabilities that currently exist but are not widely used. These non-software gaps are discussed in the appropriate context in section 2.

4. CONCLUSIONS

The focus of this paper has been on gaps in the current state of the art in manufacturing enterprise architectures, and enabling software architectures.

To fill these gaps, it will be necessary to understand the dynamic, unpredictable nature of evolving customer requirements for new products as an outgrowth in technological developments. This requirements focus will in turn necessitate the establishment of new organizational operational paradigms for manufacturing enterprises.

Mass customization, or the ability to instantaneously and seamlessly adapt existing capabilities, processes and products to meet customer requirements of arbitrary lot size, will necessarily guide the evolution of these new architectural paradigms. Manufacturing enterprises will need to imbue their organizations and processes with a culture of flexibility to meet the challenges offered by the mass customization business environment. Flexibility will be necessary both for mass customization of products and for the timely insertion of new technologies as they evolve to support manufacturing these new products.

The process flow in a mass customization must change from a linear, "waterfall" model to one that supports concurrent engineering. The software tools that support manufacturing enterprises today must be integrated. The software tools that support manufacturing enterprises tomorrow must be truly interoperable.

A computational infrastructure must be established that will support interoperability, flexibility and technology insertion. Gaps in platform availability and scalability of software tools for manufacturing can be quickly filled. Linkages across software tools will be the next step in the evolution of computational infrastructures in manufacturing enterprises. The ultimate goal is the transition to a virtual enterprise of truly interoperable software tools that concurrently support all aspects of manufacturing across geographically distributed sites of a manufacturing enterprise.

Standards activities that are currently under way provide the promise of establishing the foundations for inserting intelligence and flexibility throughout a manufacturing enterprise, including within the products that the enterprise manufactures.

About the Author

Richard W. McHard has been actively involved in requirements specification, design, development and integration of software-intensive systems for 26 years. His activities have included definition of software standards, computational infrastructures, process improvement, real-time system management, integration planning and coordination, human factors and technical management. He has performed and published research in the field of software reuse. He has an M.S. in Mathematics, with emphasis on statistics, from the University of Michigan, and an M.S. in Computer Science with postgraduate study in the foundations of computer science from the University of Southern California.

References

[1] W.Edwards Deming, "Out of Crisis", MIT CAES, Cambridge, Mass, 1986

[2] J. M. Juran, "Juran on Quality by Design", Free Press, New York, NY, 1992

[3] B. Joseph Pine II, Bart Victor, and Andrew C. Boynton, "Making Mass Customization Work", Harvard Business Review, September-October 1993, pp. 108-119.

[4] Donald C. Gause and Gerald M. Weinberg, "Exploring Requirements: Quality Before Design", Dorset House Publishing Company, New York NY, 1989

[5] Grady Booch

[6] J. A. Zachman, "A Framework for Information Systems Architecture", IBM Systems Journal, Volume 26, No. 3, 1987

[7] J. F. Sowa and J. A. Zachman, "Extending and Formalizing the Framework for Information Systems Architecture", IBM Systems Journal, 31(3), 590-616, 1992 (Zachman)

[8] ESPRIT Consortium AMICE (Eds.), "Open System Architecture for CIM", ESPRIT Research Reports, Project 688, AMICE, Volume 1, Springer-Verlag

[9] "The New Manufacturing Enterprise Wheel", SME/CASA

[10] Gideon Halevi, "The Role of Computers in Manufacturing Processes"

[11] A.-W. Scheer, "Computer Integrated Manufacturing: Towards the Factory of the Future", Second, Revised and Enlarged Edition, Springer-Verlag

[12] Robert M. Thacker, "A New CIM Model: A Blueprint for the Computer Integrated Manufacturing Enterprise"

[13] STEP\PDES ISO 10303 Documents

Chapter 4
Enabling Manufacturing Enterprise Integration

William C. Burkett
Product Data Integration Technologies
3780 Kilroy Airport Way #430
Long Beach, California 90806
310/427-3711

Abstract

Enterprise Integration is the major manufacturing challenge of the 1990's. Integration is hindered, in part, by the development and deployment of systems developed in isolation of one another that were not intended to interoperate. An Integration Maturity Framework is proposed that enables the categorization of an enterprise at different levels of integration maturity based on system interoperability characteristics. Requirements and obstacles of enterprise integration are discussed and solutions and software tools for evolution from one level of the Framework to the next are identified.

Keywords

CIM; Data Communication; Data Exchange; Enterprise Integration; Integration; Integration Maturity; Interoperability; Maturity Framework.

1. Introduction

As each functional component of a manufacturing enterprise (i.e., a "department") strives to improve its performance (thereby becoming more productive and efficient in meeting the enterprise goals), it typically turns to computers and automation to make the operation faster, more streamlined, more accurate, or more consistent. (Or they may employ automation just to keep up with the competition.) This creates a market for automation products and spawns internal application system development. To meet this market need, automation systems are developed by independent vendors for the department seeking to improve its operation. Similarly, departments for which a commercial product is not available will undertake the development of a system to meet the automation need.

The benefits of automation to each department are readily apparent. However, as the introduction of automation continues, the enterprise as a whole soon realizes that the real benefit of automation can be achieved only by integrating the different automated systems of each department. In effect, the enterprise would like to "plug together" the automated systems much like a component stereo system is assembled.

Unfortunately, integrating application systems is much more difficult than integrating stereo equipment. Each system - developed independently - makes use of its own hardware configuration, system architecture and unique data format. While benefitting the individual users, the installation of these systems leads to the aphoristic "islands of automation", to the detriment of the enterprise as a whole.

The purpose of this chapter is to (a) explain and structure this problem by presenting integration as a series of evolutionary stages, (b) identify obstacles along the evolutionary path, and (c) describe potential software solutions to overcome the obstacles and promote/enable integration.

The primary vehicle for explaining and structuring the integration problem will be by paradigm adaptation. In his book "Managing the Software Process", Ref. [3], Watts S. Humphrey presents a framework for assessing and understanding the maturity of organizations with respect to the software development process. He outlines five levels of software process maturity (i.e., how mature is the organization with respect to software development) and describes the characteristics of organizations at each level; he also explains what can be done to evolve from one level to another. In a fashion similar to the objective of this chapter, Humphrey based his framework on Crosby's Quality Management Maturity Grid, Ref. [1], which outlined five levels of quality management within an enterprise.

An Integration Maturity Framework will be presented. The Integration Maturity Framework provides a mechanism for assessing the current state of integration of an enterprise and identifies actions that must be taken to progress from one level to the next. It is within this framework that software tools will be identified and described that could support the integration evolution actions and overcome integration obstacles.

1.1 The Integration Problem

Simply put, the integration problem is that independently developed automation systems cannot effectively communicate with one another. The actual exchange of data/messages is largely a solved problem; network protocols (e.g., the OSI protocols, TCP/IP) and data standards (e.g., ASCII) exist that enable heterogeneous applications and hardware to transmit and receive digital data. The design and installation of the physical communication hardware and software is a engineering problem that will not be addressed by this paper.

So what is the problem hindering the integration of automation systems? It can be summarized in the following statements:

a) Automated systems that are developed independently of one another cannot effectively communicate.

b) Effectiveness is the measure of utility to the receiving system of the information exchanged in a communication event.

c) Utility is limited by the completeness and semantic precision of the digital information exchanged between the systems.

The reason that automated systems cannot communicate despite the physical ability to exchange data is that the meaning and interpretation of the message differs between the sender and receiver. The difficulty is in communicating meaningful information between automation systems. "Meaningful", in the sense used here, means usable by automation systems rather than by humans; people are (usually) very good at interpreting data that is presented to them, unlike software applications.

1.2 Integration Solutions

Integration solutions can be classified into three categories. These categories represent evolutionary phases of system integration and describe the principle focus of the behavior of the system. They are:

Basic Performance. The installation of network technology to provide fundamental data exchange capabilities between integrated system components. The hardware/software that provides the interconnection represents the information distribution infrastructure of the enterprise.

Effective Performance. The components of the integrated system have a common understanding of the data that is exchanged via the information distribution infrastructure. The messages sent between systems have a commonly understood meaning that enables the system to communicate.

Optimized Performance. The communication between systems is managed to provide real-time optimal performance and evolutionary adaptation as component systems and usage profiles change. The integrated system not only manages the data flow traffic and location/translation of requested information, but also can adapt the data storage configurations and traffic patterns to changing integrated system components (i.e., new systems) and changing uses of the system.

As noted above, network communications technology makes Basic Performance largely a solved problem. Systems can exchange data. The challenge with respect to Basic Performance is the adequacy of the installed system to handle the communications traffic.

Effective Performance requires a system development paradigm shift from technological interconnectivity to Semantic Integration. Integration is a communication problem; unless a commonly understood "language" is developed that the systems use to communicate/interoperate, effective communication is unpredictable.

Following the adage "it is alot easier to make working code efficient than to make efficient code work", the Optimized Performance level again introduces a paradigm shift; this time the shift is to improving the performance of the system through metrics, measurement, and analysis. Once the automation systems are integrated with respect to semantics, the systems are able to communicate. How well they communicate turns "integration" back into a technical problem. The challenge at this performance level is the identification of the rules or design principles for optimizing the performance of an integrated system.

These three phases represent an abstraction of the Integration Maturity Framework present in section 2.1.2. The details of each of these phases will be presented within the context of the framework explanation.

2. Integration State-of-the-Art

The "state-of-the-art" in manufacturing systems automation is difficult to assess because integration is not a "thing" in the sense that a CAD system is a thing, or NC milling machine is a thing. Computer-Integrated Manufacturing (CIM) literature often makes this error; instead of looking at how the individual components are put together, the literature focuses on the components that "go into" CIM. Geometric modelling, computer-aided process planning, Flexible Machining Cells, and automated NC toolpath generation are all potential elements of a CIM solution but do not, themselves, constitute "CIM".

"Integration" is a systems engineering problem, not a technology problem. The state-of-the-art in integration has more to do with system engineering methodologies than with technology.

This section will present three systems engineering approaches to manufacturing enterprise integration. Two approaches are presented in extant integration literature. The third - the Integration Maturity Framework - is introduced here.

2.1 The Integration Maturity Framework

Humphrey's Software Process Maturity framework and Crosby's Quality Management Maturity Grid represent a very powerful analysis and planning concept. By recognizing the evolutionary and "organic" nature of organizations, they decompose the subject of their work into growth stages that are characterized by certain behaviors and growth directions. Human beings mature and exhibit these stages: infant, child, adolescent, young adult, adult, mature, elderly; certain characteristics of humans change in predictable patterns as an individual ages, such as language use. In a similar fashion, organizations also mature through stages and characteristics of the organization also change as it matures.

The following sections use adapt Humphrey's/Crosby's approach to propose a mechanism for analyzing the state of system integration within an enterprise and proposing actions and tools needed to progress toward a truly integrated system.. An Integration Maturity framework is presented below that identifies five stages of integration maturity, describes the behaviors that characterizes organizations at that level of maturity, and presents actions and tools needed to evolve to the next level of maturity.

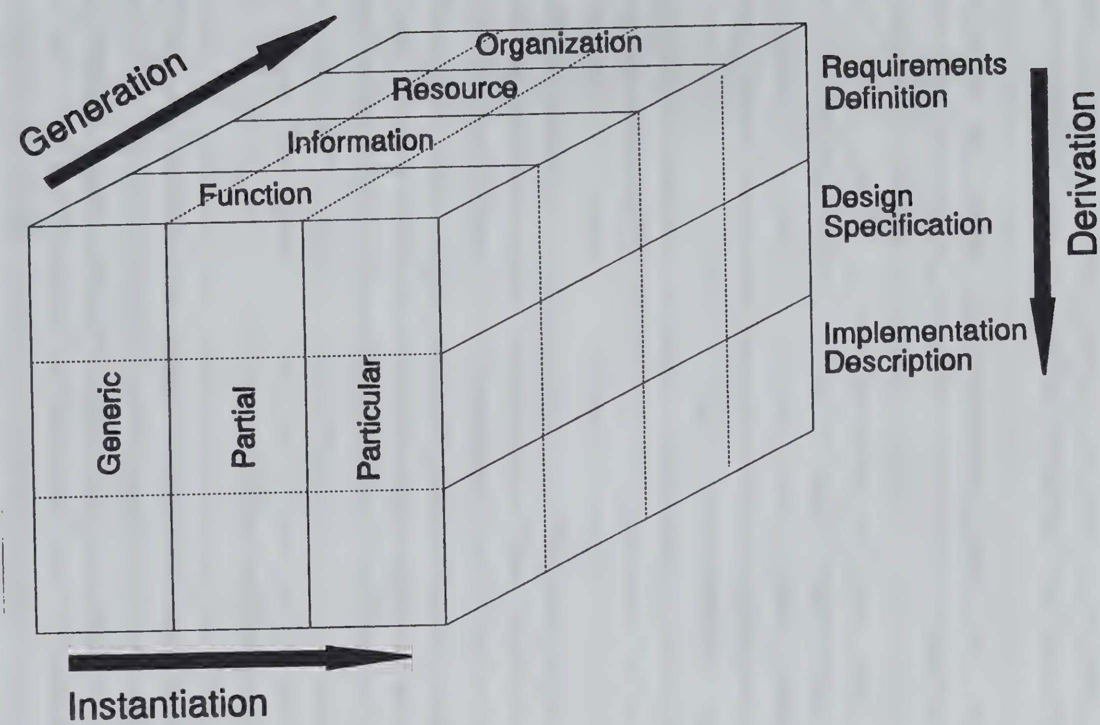
The theme of this monograph is "software gaps in manufacturing". In keeping with this theme, software tools that can aid or automate the integration process at each level of integration maturity will be identified and described. Humphrey recognizes that tools are an important part of the software process, but they must be the right tools, introduced at the right level of maturity, and inserted into the process in the right way, otherwise they can be more disruptive than valuable.

2.1.1 The CIM-OSA Reference Architecture

CIM-OSA, Ref. [2], a product of the European ESPRIT0[1] consortium's AMICE[2] project, is one "state-of-art" systems perspective in enterprise integration. The "O-S-A" stands for Open System Architecture; CIM-OSA is as a collection of reference models that structure the integration problem in such a way that integration can be planned and implemented. However, CIM-OSA takes a long-term planning, design, and development perspective on the integration problem and presents an idealized development scenario for integrated systems. It does not directly provide an understanding of the current state of an enterprise's integrated systems or recommendations on how an enterprise may progress incrementally toward an integration solution (though it does present an description of CIM evolution over time; see section 2.1.2, below).

The CIM-OSA framework is collection of interrelated system models that provides a number of different perspectives on an integrated system for the planning, development, and understanding of the system. Figure 4.1 is an illustration of the CIM-OSA Reference Architecture. Each axis of the architecture represents

1 ESPRIT: European Strategic Programme for Research and Development in Information Technology
2 AMICE: European Computer Integrated Manufacturing Architecture (inverted acronym).


Generation
Organization
Resource
Information
Function
Requirements Definition
Design Specification
Implementation Description
Derivation
Generic
Partial
Particular
Instantiation

a different set of related views of the integrated system; each cell in the cube is a systems model representing the integrated system from a particular intersection of several perspectives.

One set of views (i.e., one axis of the cubic framework) is the topical views of the system: process, information, resource, and organization. Each of these models the integrated system from a particular perspective, e.g., process model, information model. This is termed the "Generation" axis - each successive model should be generated from the previous: organization comes from resources, resources from information, information from processes.

Another set are the system (or "derivation") views and models: requirements, design, and implementation. The integrated system implementation model is derived from the integrated system design model, which itself was derived from the integrated system requirements model. This perspective and set of models should be familiar to all system developers.

The final axis of the CIM-OSA architecture are the "instantiation" views - the generalization/specialization view of the system: global view, industry view, enterprise view. For true enterprise system integration (both intra- and inter-enterprise integration), system design and development should stem from the same source so that the systems "speak the same language". A particular industry view is instantiated from a global, more abstract model; an enterprise model is instantiated from an industry model. As a whole, the framework provides a basis for information system design which, if adopted, provides a common basis for the development of inherently compatible - thus "open" - systems.

Zachman and Sowa, Ref. [4], present an information system architecture that corresponds very closely to the first two axes of the CIM-OSA Reference Architecture. The Information System Architecture they present identifies five levels of the "derivation" axis; there is an additional "strategic" perspective at the top of the axis (above "requirements") and a lower "contractor" perspective that deals with detailed requirements for contracted software. They also present six perspectives of the "generation" axis, though they assert that there is no "generation" involved; rather, all the perspectives are equally as valid a view of the system and that none takes precedence over the other; the additional views are the "event" and "motivation" views, the when and the why that complement the how (process), what (information), where (~resource), and who (organization) of the CIM-OSA architecture.

2.1.2 The Integration Maturity Framework

Although the Software Process Maturity framework is used as a basis or paradigm for the Integration Maturity framework, there are some fundamental differences between the subjects addressed by the frameworks. The Software Process Maturity framework focuses on the dynamics of organizational processes and how they change over time. The focus of the Integration Maturity framework is on the state of integration - the mechanisms and procedures used to communicate information and exchange data between automated systems - and how they change over time.

Integration Maturity Levels

Following Humphrey's/Crosby's analysis, five levels of integration maturity have been identified. These levels are illustrated in Figure 4.2.

Islands Enterprises at this level of Integration Maturity are representative of most organizations using automation in the 1970's and '80's. It is characterized by localized acquisition and point-solution application of automation technology. Systems cannot interoperate because they were not intended to interoperate.

Interfaced When an enterprise institutes standards for interconnecting applications, it has started to gain control of the enterprise system communication requirements. When applications that need to communicate can effectively communicate on a point-to-point basis using the standard interfacing approach, then the enterprise has reached the second level of integration maturity.

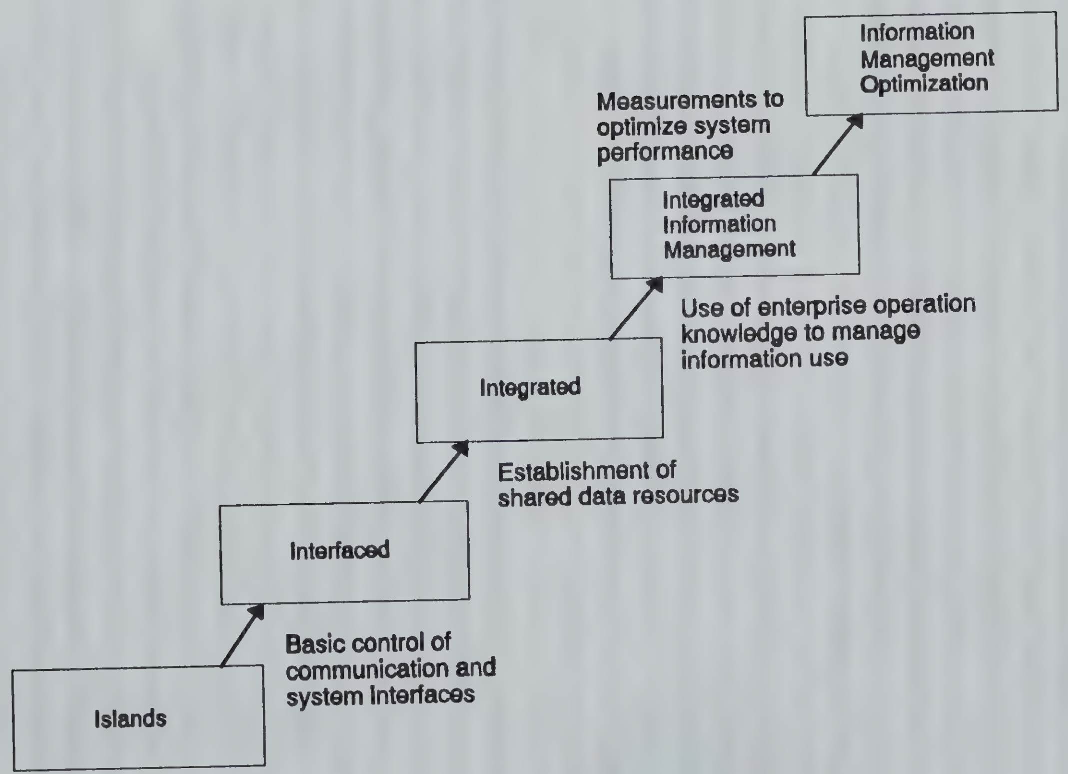
Integrated For N applications that need to communicate with one another, point-to-point links result in N * (N-1) interfaces. At the Integrated level of maturity, shared data resources are used to enable application interoperability, and resulting in 2N interfaces. While the system and information management is still primarily under user control, it is at the Integrated level real interoperability is enabled.

Integrated Information Management Once the applications are communicating effectively, the next hurdle is to provide the integrated system with advanced information management capabilities. It is at this level where the operation of the enterprise at large begins to directly affect the design and performance of the integrated system. Information/data management is under system control and the control strategies "know about" and complement the enterprise operations.

Information Management Optimization Following Humphrey's model, at the fifth level of integration maturity is a "self-aware" integrated system that can observe and measure system performance and respond to changing requirements. At this level, system integration is mostly taken for granted; the system designers/developers focus on system improvement and optimization, tuning the system itself rather than just making it work.

A complete explanation of these levels is presented in subsequent sections.

The correspondence with Humphrey's model is clear. At lower levels, the emphasis is on gaining control of the system and making it work consistently and predictably. At higher levels, measurements are established and used to monitor and improve the overall integrated system performance. In each case, the highest level is about continual improvement.


Information
Management
Optimization
Measurements to
optimize system
performance
Integrated
Information
Management
Use of enterprise operation
knowledge to manage
information use
Integrated
Establishment of
shared data resources
Interfaced
Basic control of
communication and
system interfaces
Islands

CIM-OSA CIM Evolution and Integration Maturity

The CIM-OSA research report presents a brief description and illustration of how CIM and Enterprise Integration evolves over time. (See Figure 4.3, from [2], pg. 13.) The graph presents three levels of integration that correspond to the central focus of integration efforts at a particular stage of evolution. The first level is the Physical System Integration which, as the name implies, focuses on provides the hardware and software solutions for interconnecting systems; this level corresponds roughly to the first, second, some aspect of the third levels of the Integration Maturity framework.

The second CIM-OSA level is the Application Integration, which focuses providing applications with the ability to interact and communicate; this level corresponds to the third and some of the fourth levels of the Integration Maturity framework. The third CIM-OSA level is the Business Integration which focuses on maximizing the value that can be derived from the integrated system in terms of capabilities that are provided to the user and the optimization of the system design and maintenance process; this level corresponds roughly to the fourth and fifth levels of the Integration Maturity Framework.

The Integration Maturity framework can be considered an extension or complement of the CIM-OSA Enterprise Integration Levels. The perspective of each is slightly different: the Integration Maturity framework looks at characteristics of the entire enterprise with respect to the integrated system and its use; the CIM-OSA Enterprise Integration Levels looks at the characteristics of the integrated system with respect to its dominant functionality. The CIM-OSA Levels are encompassed by the Integration Maturity levels.

2.1.3 State-of-the-Art Integration Software

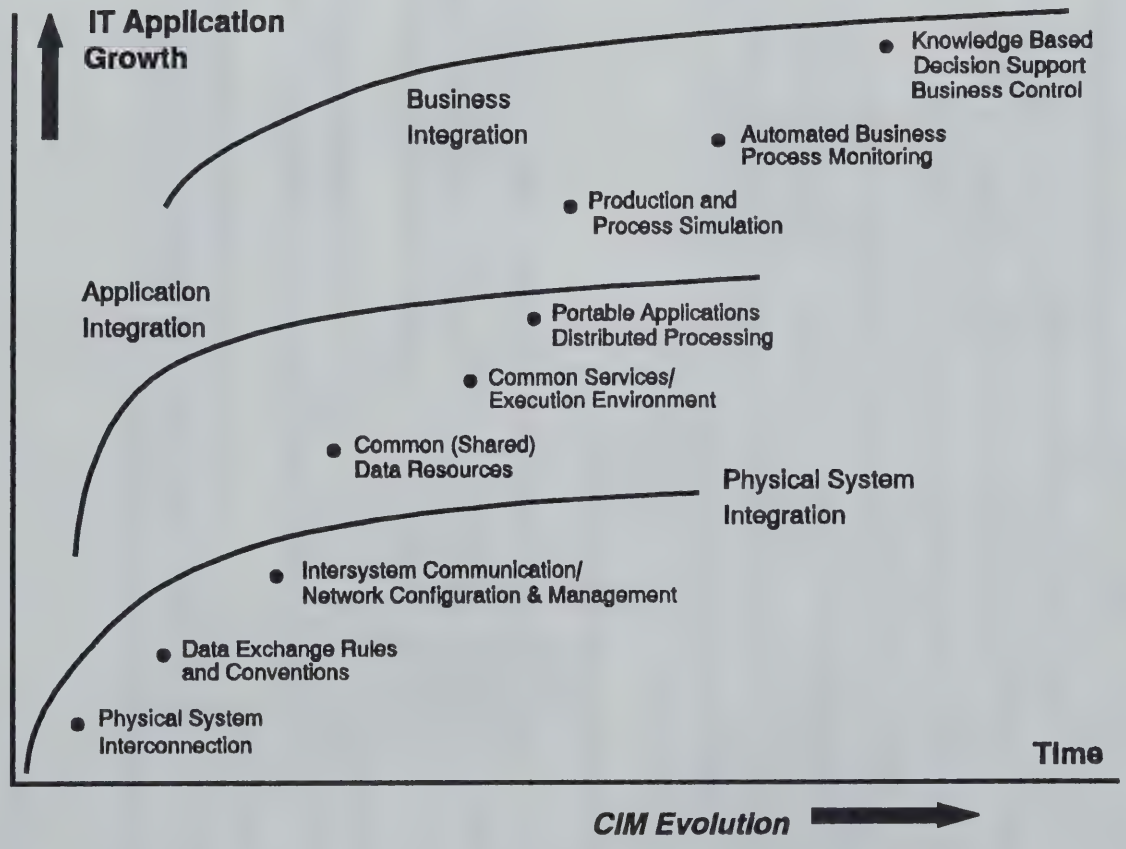
Because "integration" is a system engineering process rather than a functional discipline within an enterprise, the term "Integration Software" is open to many different interpretations. One could argue that network traffic management software is "integration" software; however, that interpretation is a little misleading because it suggests that integration software is only the software that helps/enables system to exchange data.

"Integration software" is actually a broader class of software that encompasses design and analysis tools (as well as operational tools like the traffic manager). These tools are used to perform integration.

This chapter will identify the software that supports both the operation of the integrated system and the integration process. the emphasis will be on the latter types of tools, however, since those are the tools that support progression from one level of integration maturity to the next.

2.2 Level 1: Islands

The Islands Level of integration maturity is the state of most industrial enterprises through the 1970's and early 80's. This level is characterized by the use of advanced automation technology on a narrow, localized basis; technology is developed or acquired as


IT Application Growth
Business Integration
Knowledge Based Decision Support Business Control
Automated Business Process Monitoring
Production and Process Simulation
Application Integration
Portable Applications Distributed Processing
Common Services/ Execution Environment
Common (Shared) Data Resources
Physical System Integration
Intersystem Communication/ Network Configuration & Management
Data Exchange Rules and Conventions
Physical System Interconnection
Time
CIM Evolution

point-solutions to improve functional performance within a single department. There is no overall technology acquisition or development plan, management doesn't understand advanced technology or how it relates to enterprise operations, and there is a strong, narrow, self-centered focus within each functional department with respect to both operations and system development/acquisition.

Applications are standalone and autonomous. They may be very powerful and easily meet and surpass a user's requirements, but most applications are not developed with interoperability or openness as a major design requirement (if it was a requirement at all). The applications own their databases and the databases are tightly bound to the application.

Because there are few, if any, digital communication links, day-to-day operations involve extensive re-entry of data (with the attendant risk of the introduction of human error). If links do exist, digital intersystem communication doesn't work well. There may be a recognition that system integration and digital data exchange would enable downstream applications to operate more effectively, but the only mechanisms available are 1) commercial data exchange translators that may be part of the application systems or 2) specially developed point-to-point translators. The translators convert data from one format to another - they are not application-to-application interfaces.

Digital data exchange is a rudimentary and exceptional activity; it is not part of the normal business operation. If it does happen, the interface is specialized, i.e., the mechanism and semantics are unique to the two applications/departments exchanging data. This kind of data exchange characterizes the Islands level rather than the Interface level because data exchange mechanisms and approaches are not standardized, it is not part of the everyday operation of the enterprise, and links are point solutions (i.e., unique both in approach and semantics).

The management of technology at the Islands level is very localized within individual departments and the majority of the resource expenditures are on maintaining individual applications - little is spent on building interfaces and less is spent on broader integration planning.

The characteristics of the processes that involve intersystem communication at the Islands level resemble those in Humphrey's Software Process Maturity model at level 1: chaotic, uncontrolled, unpredictable. Verbal and written communication between humans is used extensively to compensate for deficiencies of the data exchange system.

If training and standards are defined and available, they aren't used, aren't enforced, and probably are incomplete and out of date as soon as they are published. Since little value is derived from the training/standards in this state, little effort is devoted to improving them.

Enterprise Characteristics at Level 1

For each maturity level, a list of characteristics or the enterprise have been identified. These characteristics are broadly grouped into Technology, Process, and Managements categories and describe aspects of the enterprise dealing with daily operations, the state of technology, system planning and implementation, and support for integration-oriented activities.

Technology

The characteristics enumerated here are presented from the point of view of the technology that comprises the integrated system and is used by the enterprise. In other words, what are the hardware and software configuration characteristics of the integrated system, how does the configuration affect the use of the system, and what are the management policies and guidelines with respect to the system.

- No links or pathways (e.g., networks) to exchange data; no common or agreed upon linking mechanism

- rudimentary data exchange relies on commercial translators

- Standalone systems with bound, proprietary databases

- Data exchange mechanism infrequently used and of suspect quality

- The amount of resources expended on the act of exchanging data - correcting data, moving files, building specialized translators - far, far exceed the expenditures on integration planning at a broader, more comprehensive level

- Recognition that a communication link is needed occurs when the communication link is needed

- Technology is "patched in" with special purpose hardware/software - if it is linked up at all

- Little or no technology standards (i.e., enterprise-wide system requirements) beyond unused MIS recommendations

Process

The characteristics enumerated here are presented from the point of view of the processes used by the enterprise. In other words, what are the process characteristics of how the enterprise conducts its business (both operational and technology acquisition) that deal with the integrated system. For example, does the system permit, foster, or enable concurrent design?

- Process not well-defined; departments know the job they need to accomplish, but the introduction of technology either disrupts the work flow or introduces confusion over how to do the job

- Rigid process execution dependencies - one process must be complete before another is initiated

- Little visibility or understanding of what is going on in other departments

- Extensive re-entry of data

- Operations seem to be more chaotic than managed

- Alot of interdepartmental crosstalk to compensate for poor communication in other forms, e.g., "what is this file you sent me?"; primary focus of verbal communication is to clarify other forms of communication; all of this communication is non-value-added work

- Data unknowingly misinterpreted, resulting in costly expenditures based on wrong information (e.g., mirrored parts)

- Telephone and verbal communication primary means of accessing information

- Automation acquired and installed by individual functional departments based solely on localized requirements (no inter-departmental technology plan or consideration of integration with related functions)

- No coordination during technology acquisition and little during installation

<u>Management</u>

The characteristics enumerated here are presented from the point of view of the management and organization of the enterprise. The characteristics describe the relationship of the management and organization to the integrated system, its development and use. It also describes the view that the management has of the integrated system and the policies or measures that it puts in place to foster or enable integration to place.

- "That's mine!" mentality with respect to systems; sharp demarcation between departments with respect to ownership

- Upper management doesn't understand technology and relegates acquisition/installation to lower levels

- Close and pronounced correspondence between organizational structure and technology installation description, e.g., "Engineering uses CATIA and manufacturing use ComputerVision."

- Net effect of technology acquisition process is that wall are built between departments

- No control over communication/data exchange; each communication event is unique and is handled by the users communicating

- No overall technology deployment or insertion plan

- Documentation/models of what automation systems are owned by the enterprise, who is using them and for what purposes are they being used either does not exist, is not used, or is not part of an overall enterprise integration plan

- Fascination with technology for the sake of technology; technology acquisition meant to impress

- No configuration control; if any record of owned technology exists, it just an inventory (and probably inaccurate and/or incomplete)

- Upper management does not actively support MIS planning and recommendations.

2.3 Level 2: Interfaced

At the Interfaced level of maturity, the enterprise has control of intersystem communication. The exchange of data is consistent and uniform; all links between systems are created according to a standard interfacing mechanism or approach. Functional departments that use the same information have established regular communication pathways and can exchange data with a high degree of success. Exchange of data is an everyday part of the operation.

The interfaces at this level, however, are still point-to-point solutions that involve user-initiated sending/receiving of data files. Although a standard communication mechanism is used (e.g., data exchange standards, networks), each link is still unique from a semantic standpoint - there is no global, enterprise-wide definition of the data resources. The exchange of data is based on individual application-to-application interfaces, i.e., the preparation and transmission of data from a specific application to another. It also requires that data exchange be addressed on a link-by-link basis, resulting in the consideration of $N * (N-1)$ potential interfaces.

From a process and management perspective, it is still business as usual, though operations are running better. Standards are in place for development of interfaces, but there still is no comprehensive technology plan that address integration issues in the broad. The management is focused on solving of the immediate problem of getting the applications "talking to one another" rather than looking at the long term system compatibility issues, such as acquisition and development standards and enterprise-wide data definition.

With respect to the processes, the Interfaced level provides bridges between the islands. This means that at a localized level (i.e., on a given "island"), things aren't much different than at level 1. There is still a local view with respect to technology acquisition and process design; technology can dramatically affect the processes within a functional area, but has little affect at the broader enterprise level. There is still little visibility or little knowledge about "what goes on next door" and, consequently, little besides the local requirements are addressed by the new technology.

Enterprise Characteristics at Level 2

Technology

- Communication links are network or neutral file exchange; interface mechanisms are standardized.
- Emphasis on building individual bridges, establishing point-to-point communication; links are specialized mechanisms that have unique identity, configuration, and behavior
- System doesn't "know" where data is; user must find and acquire data
- Data management at local level; no on-line global data management facilities/capabilities; data management a human task and user specific
- Communication act initiated by user
- Intersystem communication is predictably successful; causes of unsuccessful communication readily identifiable and remedied
- Extra-enterprise communication possible, though problematic
- Planning vision is limited to and interoperability enabled on a point-to-point basis
- Rudimentary system design/development tools available
- Acquired or developed technology adheres to initial technology requirements standards

Process

- Functional departments have understanding of how related departments work and of their information requirements
- Communication works smoothly and predictably, though data resource managed by user

- Communication initiated and executed by user as discrete events

- Users know who data should be delivered to and where to get needed data

- Semantics of data unique to communication link

- Introduction of new technology disrupts enterprise operation; results of technology introduction unpredictable (except for the flurry of activity and confusion surrounding the introduction)

- Standards used to develop communication links

- Technology acquisition and development adheres to standards

- Basic guidelines are established for system design and development, acquisition, installation

- Standard, general purpose tools used to facilitate the development of interfaces and analyze problems/performance of the interface

<u>Management</u>

- Technology "bookkeeper" is assigned who inventories and tracks technology acquisition and use; at this level, the role of the bookkeeper is monitoring the technology and communication

- Large, specialization organizational unit devoted to development and maintenance of interfaces

- Management apprised of intersystem communication performance; management at least knows the degree of success/problems with digital data exchange

- Rudimentary functional, information, and organization models developed

- Departments still "do their own thing" with respect to technology acquisition, but management guidelines/policies are in place to make them aware of global requirements

- Management ensures that standards are used in the development of interfaces and technology acquisition

- Initial training plan formulated; training on interface design offered

- Basic toolkit established for construction of is

- o Initial management policy on integration issue - set vision and "openness" objective
- o Preliminary standards established for technology acquisition, interface design and construction, and system development

2.4 Level 3: Integrated

It is at the Integrated level of maturity that the "island-ness" begins to dissolve. While there are still standardized system interfacing mechanisms, the focus is not on point-to-point communication but on enterprise-wide communication capabilities based on a common language. This common language is the enterprise's data resource definition that specifies the information resources of the enterprise. Rather than address the communication requirements between two applications in isolation from others, comprehensive, enterprise-wide information requirements form the basis of the integrated system. The result is that rather than N * (N-1) interfaces, only 2N interfaces are needed.

At this level, there is a shift from individual applications to collections of closely interrelated, interconnected, interdependent applications. As they are designed or acquired, "workgroup" applications are not only integrated by shared data, but also by coordinated and complementary functionality - they "work together". The members of the workgroup become members either because they were designed/developed to participate in the workgroup or they are commercial software that meets membership standards set by the enterprise.

Despite the integration, however, the individual applications are still autonomous and still "own" their own data (and possibly store it in a proprietary format). The difference between the applications at the Integrated and Interfaced levels is that either:

- o the application does not own its own data;
- o the application provides a system-controlled access mechanism for retrieving and manipulating the data owned by the application.

In both cases, the database used by the application is essentially "open" to the entire integrated system. The key requirement at this level is the enterprise-wide understanding of the data available and mechanisms available access it. The integrated system software "understands" the localized data and can translate it (if necessary) into a global format for exchange. The emphasis at this level is on application-to-application communication based on a common (data) language.

The "intelligence" within the integrated system itself is still limited; the integrated system "knows" how to access data, but has little other control over the data. The user must know what data is needed and where to get it. The integrated system is at a "bit-bucket" level of operation.

This level of integration is represented by federated databases, though there are other physical implementations that can supply the same logical functionality.

At the Integrated level of maturity, management is directly addressing the integration and interoperability problem by providing indirect support in terms of high-level policies and vision, and direct support in terms of standards and training. Management actively watches the intersystem communication and manages technology acquisition and use.

The active arm of management at this level is Integration Engineering Group. The IEG is an organizational unit that evolves from the "bookkeeper" role assigned at earlier levels. The responsibility of the IEG builds on that of the bookkeeper: assess technology usage and requirements and provide support needed to enable systems to work together. The IEG is responsible for development and deployment of integration/interoperability standards, monitoring, reporting, and (helping in the) resolving of interoperability problems.

Technology acquisition does not impact the operations of the enterprise as much at the Integrated level as at lower levels of maturity. Instead of disrupting processes, the understanding of the processes and technology provided by the models enables the IEG to assess the impact of new technology on the operations and plan accordingly. The introduction of a super-duper solid model analysis system that does stress, weight, and aerodynamic analysis can be planned and coordinated with affected departments, the processes can be redesigned, the replaced equipment earmarked for retirement, and installation scheduled because there is an understanding of how the new technology affects the existing operations. It's still alot of work, but does not cause the disruption that occurs at lower levels.

Enterprise Characteristics at Level 3

Technology

- o Applications linked by logical/virtual network(s); basic Information Distribution Infrastructure installed
- o Applications more interdependent, though still mostly autonomous
- o Communication software limited to data traffic control and protocol converters
- o Applications still own their databases, though the database is treated as a repository and is accessible to/by other applications
- o Applications are interfaced with integrated system rather than other applications
- o Extra-enterprise communication through neutral file exchange
- o Users have to know where to get and put data
- o Planning vision is on enabling global communication

- Training is available on technology integration requirements, tools and methodologies, enterprise integration plans and status

- Technology standards defined for system acquisition and development; emphasize performance requirements with respect to integrated enterprise systems

Process

- Process requirements and technology capabilities are balanced - there are trade-offs; process requirements influence technology acquisition and technology capabilities influences process design

- Integration starts to break down departmental barriers; less "us/them" mentality

- Processes are able to execute concurrently; things are working smoothly - process is well-defined and understood

- User must know where data is and how to access it, where to put it

- Concurrency enabled by integrated technology, but "old way of doing business" still governs most enterprise processes

- Enterprise-wide data resources standard specified; internal communication is achieved through shared information repositories

- The impact of new technology acquisitions can be assessed and deployment planned

- Departmental technology acquisition guided by enterprise technology plan

- Training is introduced on: system use and configuration; how to get data, what is available; system acquisition; tool usage; system installation

- System design tools and methods related to integration are available and used

Management

- Integration Engineering Group formed; supports technology acquisition, installation and integration

- Management/interdepartmental communication happens regularly and smoothly, though departmental boundaries still exist.

- Management understand integration and interoperability issues through monitoring and feedback (through IEG)

- Integration Engineering Group (IEG) monitors system performance and troubleshoots problems - IEG acts as a general resource for users of the integrated system

- Functional, information, and organizational models are developed and used for system planning and design; rudimentary event, facility, and business models developed

- Management enforces conformance to system design/acquisition standards - system must be installed with integration as a high priority requirement

- IEG issues data management standards and procedures

- Application compatibility requirements developed, standardized, published

- Training on planning and implementation initiated for system builders

- Management commitment, vision, and plan for enterprise integration published and distributed throughout enterprise

2.5 Level 4: Integrated Information Management

The difference between the Integrated Information Management level of maturity and previous levels is that there is a shift in focus with respect to what is important. Rather than emphasizing basic interoperability issues which dominated levels 2 and 3, levels 4 and 5 emphasize performance issues. Level 4 focuses on the management of information, and level 5 on the optimization of information management performance. At level 3, the system is working; at level 4, it is working better.

At level 4, the systems not only interoperates effectively, but the system itself actively promotes the interoperability. Rather than simply being a tool of the user, the integrated system understands the enterprise processes and "knows" what user/system needs what information. Routine communication shifts from user control to automatic control. Thus, when designs are released, the appropriate parties are notified of the release and/or receive the designs.

At level 3, there is system-wide access to all repositories, but the repository may still be owned by an application. At level 4, data repositories become system-owned and managed. Applications are de-coupled from sharable data (they will still need local data) and the information distribution infrastructure becomes an enterprise service; it stores, tracks, supplies, and manages the data resources of the enterprise - the data is now "owned" by the enterprise as a whole. Data management is the responsibility not of the users, but of the integrated system.

Planning, development, acquisition, and installation standards are defined, used, and enforced to ensure that the integration/interoperability requirements are part of the system evolution. No system is acquired without considering compatibility issues with the integrated system as a whole. Tools, libraries, and other support mechanisms are in place to ease system installation.

In addition, a new system function is introduced: performance measurement. Since the integrated system is working and is managed, the next step in improving the level and quality of integration is the examination of the system and how it performs. Monitoring is no longer a passive activity, but rather is actively used to control data access and distribution. This requires the specification of metrics and the instrumentation of the system to gather performance data.

Requirements for system design, analysis, and implementation tools and methods become part of the system design, analysis and implementation requirements. System building tools begin a transition from standalone, objective implements for constructing the integrated system to being part of the system itself. Although the integration of operational and system-building software is not a level 4 characteristic, the "hook" and requirements must be installed at level 4 in anticipation of level 5.

The integration of system-building tools with operational software is a companion task to the instrumentation of the integrated system for performance measurement. Among the tasks for the IEG and enterprise management at the Integrated Information Management level is the specification of performance metrics to measure and analyze the use of the integrated system. During the insertion of information management capabilities to the system, both instrumentation for measuring system performance and "hooks" for system-building tools can also be added with little additional effort.

Enterprise Characteristics at Level 4

Technology

- Data management/repositories under system control; all applications use repositories
- System Information Manager "knows" where data is, how it is stored, and how data in different repositories are related
- Applications no longer "own" data; rather the enterprise as a whole "owns" the data
- Integrated system is instrumented based on performance metrics
- Data management services mostly invisible to user
- Extra-enterprise communication is performed consistently and successfully, through networks, file exchange, translation; still a user initiated event

- Repository schema design still performed by humans, though with clear physical, syntactic and semantic requirements

- Repository schema design to accommodate new applications is straightforward and has a minimal impact on system operation

- Requirements for integration of planning and design tools are part of technology planning and acquisition requirements

- Full training on tools/methods, enterprise process education and how tools support it; training on technology performance assessment instituted

Process

- Process changes are planned and executed according to technology acquisition, product changes, and management changes

- Enterprise process includes self-analysis processes to monitor and measure performance

- Routine communication automated, e.g., design releases

- Requested information is easily found and accessible

- Tasks performed concurrently - Concurrent Engineering enabled

- Communication/data exchange/information sharing is background task - mostly transparent to the user

- Technology acquired according to technology plan; installed with little disruption

- Process understanding (models) actively used to monitor and improve the process

- Procedures are developed for evaluation of system performance (which incorporate and apply performance metrics)

- Data management procedures incorporated into integrated information system manager

Management

- Management is integration savvy; management focus on controlling data as a resource, much like inventory control of raw material and finished products

- Measurement and tracking of system performance initiated

- o Feedback to management is regular and meaningful (i.e., the value of the measurement of a performance metric provides meaningful information about the performance of the system

- o Technology evolution plan incorporates design tools and models into integrated system (enables rapid response/reconfiguration to changes)

- o Requirements for extra-enterprise communication is part of technology development plan

- o Performance metrics are specified - applications/integrated system are instrumented

- o System performance standards are specified

- o Training on integrated enterprise architecture (IEA) initiated for all IS users and builders

2.6 Level 5: Information Management Optimization

The Information Management Optimization level of Integration maturity has several prerequisites that are met by progressing through the earlier levels:

- o The system works effectively - digital intersystem communication is part of the everyday operation of the business and is consistent and successful;

- o The data resources of the enterprise are managed as an enterprise resource;

- o Standards, models and specifications exist that both guide the integrated system evolution and document the configuration and performance of the system;

- o The Information Distribution Infrastructure is an identifiable component of the integrated system and is instrumented to gather data for evaluating system performance.

It is only once these objectives have been achieved that the components are in place for real system optimization; it is only when a system is completely understood and objectively examined that concrete, measurable improvements can be introduced. At level 5, the Information Distribution Infrastructure becomes an enterprise server; the focus of improvements is the performance of the server, enabling it to provide information to the client applications more efficiently, effectively, and quickly. Ideally, the system is self-optimizing, able to reconfigure itself based on usage patterns (which reflect shifting/changing information requirements).

In fact, at level 5 there is a much greater synergy between the technology, process, and management. The impact of new technology on the existing system(s) and processes can be readily assessed, and the technology can be easily assimilated into the system. In addition, the operation of the system provides clues and requirements for new technology - no longer solutions in search of problems, but rather performance improvement opportunities in search of solutions.

The name of this level - "Information Management Optimization" - is a little misleading and little incomplete. To reach this level of maturity, there are a wide range of changes and improvements that are needed in the enterprise; the integrated system itself cannot mature alone, independent of the maturation of the enterprise as a whole. "Information management optimization" is just one small part of the overall enterprise performance optimization. It is meant to imply not only the optimization in information management, but the acquisition and development of technology, the execution of enterprise processes, and the management of technology and processes. Technology, process, management, resources, and information are all inextricably bound to one another - changes to one result in reactions by another, yielding an extremely complex interplay of system characteristics and behavior. Making the situation more complex are extra-enterprise environmental factors that impinge on the configuration of the enterprise.

As observed by Humphrey in his explanation of level 5 of the Software Process Maturity framework, "optimization" is not an end-state, but a state of continual improvement (which is why he calls it "optimizing"). The environment within which the enterprise operates will continuously change, which means the enterprise will have to change in response (or eventually perish), which means that the resources of the enterprise (e.g., the integrated system) will also undergo continual change. The self-evaluation, continual self-improvement, and managed evolution is on ongoing part of the enterprise and must be part of routine enterprise operations. Technology installation is not a destination, but merely a intermediate state in a series of (hopefully) neverending state transitions.

Decision support is a major system function at level 5. Continuous change and improvement means continuous decision making. The complexity of the integrated system (and the enterprise as a system) is extreme and beyond the ability of human planners to grasp. Therefore, automated means of evaluating the impact of system design changes and simulating system performance are necessary to plan and implement changes.

The use of decision support technology applies not only to the planning and implementation functions, but to operational processes as well. The level of "intelligence" in the integrated system is such that a wide range of decision support applications can be created that were not possible at lower levels of maturity. Process planning, for example, can take advantage of system knowledge of the scheduled machine workload and importance/need-date of the part to be machined to plan alternate routings for the part.

The tools and methods used to evaluate and implement the changes are given the same level of importance and integration as operational systems and are used as part of the routine operation of enterprise. System acquisition, development, and installation are not exceptional activities - they are valued-added activities and just as important to the functioning of the enterprise as cutting metal.

Enterprise Characteristics at Level 5

Technology

- o Applications decoupled from repositories; local data control limited to working and scratch data

- o Enterprise Integration Architecture is integrated with integrated system - system "knows" about itself

- o System performance is self-optimizing based on performance metrics and specified responses (e.g., reallocation of data and reconfiguration of repositories)

- o Extra-enterprise communication indistinguishable form internal communication

- o Acquired technology can be "plugged in" to the integrated system and engaged; little or no compatibility or customization efforts required

- o "Openness" or "integrability" is essential performance requirement for acquired systems

- o Planning and design tools, methods, and models are integrated with application/operational software - which enables rapid response to system design changes

- o Standards for technology performance, acquisition and installation are defined

- o System design and development tools and methodologies are actively used as an operational part of system maintenance, acquisition and evolution

- o Training addresses use of all technology (operational applications and tools), use of technology to support enterprise operations, and technology availability

Process

- o Planning/process definition ensures that concurrent execution is part of operation

- o Standard operational processes include self-correction, improvements, and/or optimization processes

- o Execution of processes mostly driven by automation; while humans still perform work, work flow is monitored and controlled by system; less reliance on human action

- o Rapid, planned response to change; enterprise models are embedded in the integrated system so impacts of changes can be evaluated and planned

- o Acquisition, design, development, installation, and usage standards used

- o New technology has understood role in integration framework and can be introduced and immediately employed ("plug in" and go).

- o Process standards are defined for system design, development, acquisition, installation, and use

- o Training is a routine operational activity that provides instruction on integrated system technology, configuration, and use

- o Tools are integrated with planning and implementation process

- o Training on system understanding is computer-based instruction - the system teaching users about itself.

Management

- o Organizational/management boundaries blur - strong sense of "team"

- o Management awareness and support of system integration and interoperability still exists, though the performance of the system is such that it doesn't require alot of management attention

- o Day-to-day management of communication disappears - completely internalized in system

- o Management activities focus on improvement of system

- o Performance measurements routinely made

- o Continual improvement and optimization procedures and criteria are specified and in place

- o Performance data used for planning system improvements

- o Enterprise Integrated Architecture is used for integrated system planning and acquisition

- o Standards defined, used, and enforced

- Complete training for enterprise established for users/builders/analysts with respect to the integrated system

3. Integration Gaps

The picture of "integration" described above provides a context for the identification of "integration software". At each level of integration maturity, different kinds of software tools are necessary to operate at that level and to progress to the next level of maturity. This section identifies the tools and the roles that they play in integration.

3.1 Progression from Level 1 to Level 2

As with Humphrey's model, the biggest challenge faced by an enterprise at the Islands level of maturity is getting control of the situation. While the enterprise may have a good understanding of how work proceeds, the specific details of how work is done and what technology (hardware, applications, tools) is used to do the work are unknown. As a result, the enterprise is unable to make any plans concerning either the integration of the existing technology or acquisition and installation of new technology. Therefore, one of the first steps necessary to progress from level 1 to level 2 is to identify the technology used in the enterprise.

But an inventory of the technology isn't enough. How the technology is currently used must also be documented; this suggests that the operational processes used by the enterprise must also be documented. This perspective, however, is an analysis perspective and can be considered as bottom-up. From a top-down, design perspective, it is the process specification that should drive or control the technology that is used to perform those processes. The most important first step in integrating the enterprise systems has nothing to do with technology, but rather it is understanding the processes by which the enterprise goes about its business.

There are tradeoffs, of course; a new and improved tool will often not only change the way a task is performed, but sometimes even change the task itself. The tradeoffs between technology capabilities and process design must be made on a case-by-case basis within the context of the overall process specification and integrated system configuration.

Progressing from level 1 to level 2 is more than compiling an inventory of technology or documenting processes. These thrusts correspond to two of the three perspectives used to classify the characteristics above; the third perspective - management - also offers its own unique challenges. Progressing to level 2 involves not only analysis and planning from a technology and process standpoint, but also, most importantly, overcoming organizational inertia and territorialism present in many enterprises.

In many respects, the enterprise characteristics that foster integrated system evolution are the same as those that foster quality and productivity improvements: a common vision and sense of "teamness", confidence in the management that they know what they are doing and

are taking rational action to guide the evolution of the enterprise. Each member of the enterprise must know where they fit, the role that they play, and the contribution that they make to the health and evolution of the enterprise as a whole.

The focus of this chapter, however, is on system integration and the evolution of integrated systems. The broader issues concerning healthy enterprises are acknowledged here because the contextual issues always affect the analysis and understanding of any problem. It is important to keep them in mind during the analysis of an narrower subject (such as system integration) in order to separate organizational requirements from requirements unique to the narrower subject.

3.1.1 Significant Issues and Obstacles to Overcome

Progressing from level 1 to level 2 involves overcoming a number of obstacles and making a decision with respect to a number of issues. These include:

- Ad hoc, self-centered technology acquisition

- Widespread ignorance of:
  - Functional operation of enterprise
  - Enterprise information resources
  - Systems owned and used in the enterprise
  - Enterprise organization

- Inventory and configuration management of technology

- Grasping problem

- Understanding information flow and use in enterprise

- Re-focusing perception from local to global with respect to technology acquisition

- Threat of loss of control of technology; departments will often keep technology hidden from enterprise planners out of fear that is may be taken away

All of these involve a transition from a narrow, localized view of the enterprise to a more global view of the enterprise.

3.1.2 What must be done to progress from Level 1 to Level 2

There are actually two phases in the progression from one level to the next. The first is gaining control and operation at the current level of maturity. The second is taking active steps to progress between levels. The second phase at one level overlaps with the first phase at the next level; the second phase for level 1, for example, is the first phase for level 2. The

actions and tools that involve the interfacing of applications can therefore be described at either level. Both phases will be described the current level and the second phase will then be assumed at the next higher level of maturity.

The actions required to progress to level 2 obviously must respond to the issues and obstacles listed above. Principle among these is getting a grasp on the problem by understanding how the enterprise conducts business. The creation of models of the existing processes and systems and the development of technology plans is the first step.

Phase 1 actions

- o Enterprise planning/modelling
- o Develop technology plan; long term plan for technology acquisition and insertion
- o Assign technology "bookkeeper" to inventory and track technology owned and used (or not used!) by the enterprise (this is a functional precursor to an Integration Engineering Group)
- o Inventory and control of technology (DBA kind of toolkit)
- o Remove or mitigate threat of loss of control (to the "bookkeeper"); assure departments that tools won't taken away with better replacement
- o Management focus on getting control and predictability of intersystem communication

Phase 2 actions

- o Establish interface standards and development procedures
- o Develop analysis and reporting mechanisms (based on process models) to understand intersystem communication requirements
- o Establish system acquisition standards and procedures to ensure that new systems "fit" with existing systems
- o Establish system development standards and procedures to ensure that systems are open to future integration

3.1.3 What software tools can aid progress from Level 1 to Level 2

The software tools that can best support the progression from level 1 to level 2 are, for the most part, general purpose management and analysis tools. At this level, powerful niche tools are of no value because the technology and processes are generally uncoordinated and the

relationship between them volatile and uncertain. For example, the value of a network data manager is unknown until the processes are designed, technology is specified to support the processes, the network configuration is determined.

Once again, while there are obvious tradeoffs between technology and process, caution should be exercised to prevent the technology from dictating how tasks should be performed.

Basic software tools to aid the progress from level 1 to level 2 include:

- Basic modelling tools to model characteristics of the enterprise; specifically:
  - process/functional modelling tools
  - technology/resource modelling, inventory, or management tools

- Project planning tools

These tools and descriptions of required actions are related to the first phase of progress between levels. The tools used in the second phase that involve the interfacing of applications are:

- Data browsers

- Neutral data exchange format translator front-ends

- Code generators to write interfaces

- Data mapping languages/tools or cross-reference mechanisms

Software tools, like the actions, fall into two categories: those that aid the operation of the enterprise at a given level of maturity and those that aid the progress between levels.

3.2 Progression from Level 2 to Level 3

At the interfaced level of Integration Maturity, systems that need to exchange data can exchange data with a high degree of reliability and success. Problems associated with exchanging data between systems in day-to-day operations are largely eliminated. Once communication pathways and application interoperability are established, efforts can be applied to improving their performance by installing a more sophisticated communication mechanism.

The progression to level 3 involves changing the focus and philosophy of intersystem communication. Instead of building individual bridges between the islands of automation, an Information Distribution Infrastructure is created to provide mechanism for data storage, retrieval, and transmission. This reduces the number of interfaces required from N * (N-1) to 2N, since all applications are interfaced to one another through a central data control system.

To do this requires bringing the functional departments within the enterprise closer together by defining enterprise-wide data resources. Building on the enterprise process model,

a conceptual model of the information used by the enterprise is created. The conceptual model specifies the semantics of the data contained/stored in the repositories of the integrated system (but does not specify the format of the data).

The primary technical challenge in progressing from level 2 to level 3 is "opening up" the proprietary databases that are bound to applications. True integration means that all enterprise data resources are known and available to processes/applications throughout the enterprise.

3.2.1 Significant Issues and Obstacles to Overcome

Progressing from level 2 to level 3 involves overcoming a number of obstacles and making a decision with respect to a number of issues. These include:

- o Enterprise-wide data definition
- o Definition/specification of repositories
- o Configuration management of technology
- o "De-coupling" or "opening up" application databases; (in order to perform real integration, the databases of the applications that are part of the system must be completely understood and accessible by the system, either directly or through a standardized data access interface)
- o Access to proprietary databases
- o Installation of standard communication mechanisms
- o Duplication/concurrency/redundancy of data

3.2.2 What must be done to maintain level 2 performance

As described above, there are two phases involved in the progression between levels. The first phase - actions involved in maintaining a given level - of level 2 is partially described as part of the actions required to progress from level 1 to level 2. These include:

- o Develop enterprise process model
- o Develop inventory of technology and its use (particularly within the context of the process model)
- o Enforce interface, acquisition, and development standards

3.2.3 What software tools can aid the operation at Level 2

There are several kinds of tools that aid the operation of the enterprise at level 2. One kind are tools that enable the enterprise to gain control and understanding of the processes used and the technology owned by the enterprise. Another are tools the facilitate the building of interfaces between applications. These tools include:

- o Data browsers and visualizers
- o Neutral data exchange format translator front-ends
- o Code generators to write interfaces
- o Data mapping languages/tools or cross-reference mechanisms
- o Technology inventory, management, and modelling tools
- o Process modelling tools (ideally integrated with technology model)

3.2.4 What must be done to progress from level 2 to level 3

The second phase focuses on progressing between levels. Actions required to progress from level 2 to level 3 include:

- o Move away from import/export philosophy to enterprise-wide data definition
- o Semantic Integration; design and specification of information resources and data repositories
- o Change emphasis of data usage from satisfaction of individual needs to broader perspective of enterprise information resource requirements (which reduces the instability of interfaces)

3.2.5 What software tools can aid progress from Level 2 to Level 3

Progressing from level 2 to level 3 requires tools that deal with enterprise-wide data resources and data management. Modelling tools, conflict resolution aids, data analysis tools (for reverse engineering existing repositories) and data managers are needed to progress to the Integrated level of maturity.

- o Schema designers

- o Semantic modelling languages
- o Requirements analysis tools
- o Data analysis and access tools for understanding and using existing repositories
- o Data management tools to facilitate the exchange of data through the distribution infrastructure, such as data traffic managers

At level 3, the integrated system requirements begin to make room for niche solutions. Special purpose tools like data managers and usage analyzers become more useful and applicable to system analysis and design.

In progressing to level 4, the nature of the software also begins to change. Earlier levels required general purpose design and analysis tools; level 3 introduced the requirements for system software that is an active element in the operation of the enterprise. At level 4 design and analysis tools are no longer general purpose, but are specific to the needs and configuration of the integrated system.

3.3 Progression from Level 3 to Level 4

At the Integrated level of maturity, many enterprises will feel that they have reached an ideal end-state, an automation nirvana where systems can exchange data, interoperate freely and effectively, and actually help the enterprise operations rather than get in the way or make more work. While it is true that most of the non-value-added costs are eliminated at the level 3, there are still dramatic improvements that are possible. Progression to level 4 echoes the heuristic it is easier to make working code efficient that to make efficient code work

In other words, once the integrated system is working, then it is easier and more straightforward figuring out how to make it work better.

Evolution and improvement of the integrated system beyond the third level involves two main thrusts:

Active information management by the system; and
Measurement and analysis of the operation and performance of the integrated system.

Progress from level 3 to level 4 concentrates on the first of these (the second is addressed in the progress from level 4 to 5). At level 3, the users are still responsible to intersystem communication and data storage and retrieval. The data user must know what data is needed, where to get it, and how to translate it into a native format (if needed); the data producer must know who needs it, where to store, and how to notify the user that the data is available. Progression to level 4 requires the installation of intelligent management software that take over these tasks, that actively controls data usage, access, exchange, translation, and storage.

The evolution from level 3 to level 4 illustrates that the enterprise has full understanding and control of its information resources.

In addition to data management, the integrated system understands the processes used by the enterprise and can assume responsibility for routine communication. The system would store data from data producers in/at cites where the data is most likely to be next used and notify users of data releases.

3.3.1 Significant Issues and Obstacles to Overcome

The biggest obstacle to overcome to progress to level 4 is understanding data usage patterns within the enterprise. The process models represent how the enterprise does business; the technology/resource models represent how technology supports or is used to the do business; the information models represent what is used or processed during the course of business. But models are just static representations that do no reflect the dynamics of real world systems. An information flow in a process model is simply a line and all the lines on the model look alike and suggest that all the lines equally important to the enterprise - which couldn't be further from the truth! Some of the information flows are significantly more important than others.

Directly related to this issue is the development of metrics and the instrumentation of the integrated system. While measurement and optimization characterizes the progress between levels 4 and 5, the definition of metrics and measuring practices and mechanisms must be started during the progression from level 3 to level 4. Understanding the data usage patterns requires monitoring of the actual usage; monitoring is the first part of measuring. Therefore, the mechanisms installed to monitor the intersystem communication can provide information to both actively manage the data usage and assess optimization opportunities.

Throughout level 3 and into level 4, the issue of data ownership will continue to be a problem. The integrated system must have control of the data resources and access to application databases (if the repository cannot be decoupled from the application).

Specifically, obstacles include:

- o Understanding data usage patterns
- o Local data control
- o Definition of metrics for and analysis of data usage
- o Technology evolution plan; what is the target and how are changes introduced
- o working with vendor to decouple application database

3.3.2 What must be done to maintain level 3 performance

Maintaining operations at the Integrated level of maturity involves widespread and constant consciousness of the integrated system as a single entity. Standards must be developed, refined, and used to ensure that systems that are developed or acquired fit with the existing system or can easily be adapted to the existing environment.

In addition to those needed to progress to level 3, specific to actions to maintain operation at level 3 include:

- o Definition and enforcement of integration standards

- o Management support of the IEG charter and activities

Since standards become more important and play a bigger role at each level of maturity, another evolutionary challenge is keeping abreast of national and international standards related to data exchange, intersystem communication and interoperability. Without the recognition that extra-enterprise factors exist and the visibility of industrial trends, an enterprise is in danger of "painting itself in a corner" with respect to standards and integration technology. Much like the single-vendor integration solution, solely internal integration planning is a danger to future growth and partnerships. Knowledge of and compatibility with national and international standards will leave open future opportunities for integrated system growth - which is particularly important in the era of information superhighways and electronic commerce.

3.3.3 What software tools can aid the operation at Level 3

Software tools used at level 3 are tools that promote and enable the interoperability of systems. It is not efficiency or optimization that is important, but rather simply basic functionality. The objective is to provide tools and software aids to interconnect the applications through a common infrastructure.

- o A standardized data access interface that can be applied at a global level

- o Product data management tools that perform concurrency/integrity control, check-in/check-out, and versioning

- o Reverse Engineering tools for new system acquisition and installation; tools are needed to analyze and decode the repositories of "less than open" applications

- o Libraries of standardized program modules that deal with access/interfacing repositories and the integrated system

- o Libraries of standardized objects or other representations of the enterprise data resources

3.3.4 What must be done to progress from level 3 to level 4

Many improvements are instigated and installed at lower levels of integration maturity. The use of enterprise models, standards, and training, and management commitment to an integrated enterprise must be continued throughout the evolution and existence of the integrated system.

Progressing from level 3 to level 4 requires instilling the integrated system with enterprise process and system knowledge, enabling it to actively manage information usage. Finding and retrieving needed data should no longer the be the responsibility of the user, but a responsibility of the integrated system. Not only does this improve the users ability to perform, but permits the system to make transparent changes to the underlying storage mechanisms (i.e., repositories) in response to system changes.

System design, analysis, and development also changes. Instead of the strictly objective approaches to system improvement, the tools and methods start to be part of the integrated system itself. So not only is the integrated system more "aware" of its configuration and its role in the operation of the enterprise, but it begins to possess the ability (i.e., the "hooks") to see and improve itself. The integration of design tools and methods with the integrated system takes place between levels 4 and 5; between levels 3 and 4, the requirements for the integration of the tools and methods become part of the requirements for system design, analysis, development, and acquisition. Progressing from level 3 to 4 with respect to design tools is a preparatory phase, setting the stage for the installation and integration of the tools. The objective of these requirements is to provide the system with built-in mechanisms for changing and evolving over time because static, inflexible system become obsolete almost overnight.

- o Installation of an automated System Information Manager for automated control of communication (requires complete understanding of enterprise information resources)

- o System design and acquisition requirements include requirements for design tools and methods

3.3.5 What software tools can aid progress from Level 3 to Level 4

There are two kinds of tools that enable and promote the progress from level 3 to level 4: (a) data management tools that are transparent to the users and service the interoperability needs of the integrated system; and (b) analysis and monitoring tools to begin the assessment of the system performance.

- o System software that monitors and data access and traffic

- o Information managers for the enterprise data resources

- o Monitoring/analysis tools to examine data usage patterns and requirements

At level 4 and on to level 5, the software tools to support the development and use to the integrated system become more niche-oriented.

3.4 Progression from Level 4 to Level 5

At the Integrated Information Management level, the integrated system "knows" enough about the operation of the enterprise to be a powerful motive force that enables enterprise operations to move forward swiftly, predictably, and confidently. Not only have intersystem communication problems long since disappeared, but much of the responsibility for communication, interoperability, and data management has been automated and is longer a worry of the user. There is, however, one more evolutionary step that the integrated system must undergo.

At Level 4, the enterprise has complete descriptions, specifications, or representations (i.e., models) of the:

enterprise operational processes;
system design, analysis, and implementation processes;
technology resources;
information/data resources;
system performance characteristics and metrics; and
(plan for) integrated system evolution.

In short: a complete understanding of the state of the enterprise. Progress to level 5 involves using this understanding and the measurements to identify alternative states or system changes which improve the performance of the integrated system (in particular) and the enterprise as a whole. Which is simply the engineering method: understand how something works, measure aspects of how it works, then identify alternatives to make it work better.

Progression from level 4 to level 5 means giving the integrated system the "self-awareness" to monitor how it works and the ability to reconfigure itself based on changing usage requirements. For example, if a kickoff of a new project results in a sudden increase in the number of design created, data storage resources can be reallocated or the system can request additional resources. The system understands not only how the enterprise operates, but how it operates as well. The integration of the system-building tools, procedures, and models with the operational system provides the mechanisms for the self-awareness and the ability to rapidly change this understanding - and the performance of the system itself, based on the degree to which the system performance is based on its understanding.

The major changes to the integrated system in moving from level 4 to level 5 are:

the integration of system-building tools with the operational software; changing the operational software such that it is driven by models of the enterprise rather than a hard-coded understanding of the enterprise; and gathering system performance data and using it to analyze the performance and investigate system improvements.

From a management perspective, the evolution to level 5 means a focus and emphasis on continual improvement. As at lower levels, much of this depends on team building and management practices that are good business practices, but beyond the scope of this chapter.

3.4.1 Significant Issues and Obstacles to Overcome

The major obstacle to overcome in the progression to level 5 lies in the appropriate response to the observations of system performance (i.e., measurements). If data requests for a certain repository suddenly jump, is it perturbation or a change in the requirements? Should the data be redundantly stored in different repositories to respond to the requests more quickly? These are difficult questions and the answers obviously depend on the local conditions or situation, which is why this is an issue. The metrics are not simply measurable numbers (e.g., lines of code), but must mean something with respect to system performance and a change in the number must be assessed as being good or bad with respect to system performance.

The technology, itself, may prove to be an obstacle. There are applications and there are system-building tools - is it possible to integrate them? Is it possible to create a "self-aware" system that can monitor and improve its own performance? While the answer to these is "of course", actually installing operational software with these characteristics will be a challenge.

Specifically, challenges and obstacles include:

- Understanding and specifying proper response to system changes
- Understanding relationship between the integrated enterprise system and enterprise operations
- Gathering the right data and determining appropriate response
- Establishing design rules for repository schemas; self-configuring repositories

3.4.2 What must be done to maintain level 4 performance

Maintaining performance at the Integrated Information Management level requires the recognition that the data resources used by an enterprise can be treated like other enterprise resources. Data management is subject to the same principles of material management. (Though the associated costs are radically different: the cost of data transport is virtually non-existent compared to material transport; the cost of data loss, however, may be much greater than material loss.)

Specific actions required to maintain level 4 are:

- Continuous monitoring of system behavior and modification of controls based on changes in behavior

o Planning and implementation of integrated system changes to provide system performance data and enable system-building tools to be integrated with operational systems

3.4.3 What software tools can aid the operation at Level 4

Software tools for operating at level 4 fall into two categories: those that enable intelligent management of the data (which includes monitoring software); and those that aid the analysis and modification of applications and system software (i.e., operational software). The later category of tools facilitates the instrumentation of the system, the incorporation of models into the system operation, and the integration of system-building tools into the integrated system. Specific tools include:

o tools for monitoring data usage and repository demand and performance

o intelligent data managers

o computer-based educational aids that provide instruction on system configuration and usage, data resource availability, enterprise processes, and other aspects of the enterprise (basically, educational aids based on the enterprise models)

3.4.3 What must be done to progress from level 4 to level 5

Where level 4 concentrates on information management and, concurrently, preparations for system analysis and rapid redesign, the progression to level 5 requires acting on the measurements and using the tools to optimize the performance of the integrated system. The objective is to be able to measure the system, determine what could be done to improve the system, and make the improvements as quickly as possible.
Specific actions include:

o Complete the descriptive enterprise model(s); models are used to evaluate impact of changes to: technology, operation, organization, facilities, and management policies/directives on the other aspects of integrated system;

o Characterize system behavior (i.e., enterprise model(s)) in a form understandable by the system;

o Take measurements of system performance; determine how to respond to measurements;

o Integrate system-building tools and methods with operational systems; the system-building tools create the models that
drive the system management software.

3.4.5 What software tools can aid progress from Level 4 to Level 5

The monitoring and measuring started in level 4 continues and expands its scope in level 5.

- o tools/daemons for monitoring system performance characteristics and metrics (besides usage and demand)

- o model-based operational software or tools for incorporating models into system performance

- o decision support software that suggests system design alternatives based on system measurements

3.5 Operation at Level 5

The final level of Integration Maturity is much like the last level of Humphrey's Software Process maturity: even though it is highest point of evolution doesn't mean that it is a destination. While something very significant has been achieved, the real praiseworthiness comes not from the achievement itself, but from maintaining that level of performance. And maintaining the last level of maturity may be more difficult than maturation process itself.

3.5.1 Significant Issues and Obstacles to Overcome

Just because an enterprise reaches the final maturity level doesn't necessarily mean that things get easier. Operating at the Information Management Optimization level of maturity still involves challenges and obstacles, though, like transitions between earlier levels, the nature of the challenges and obstacles change.

The primary challenge in maintaining level 5 performance is the natural tendency of systems to degrade over time. When things are functioning well, self-improvement becomes less of a priority; when major challenges are met, the urge to relax and coast sets in. The challenges at level 5 do not involve technology as much as the people that are part of the system. The ennui and entropy that follows the successful completion of a project or the overcoming of a challenge are the biggest dangers. Once level 5 is reached, there will be tendency for the people to feel a sense of completeness or that a destination has been reached.

The flip side of these challenges is that they may not ever arise because an enterprise may never reach this level of maturity. The organic complexity of an enterprise at this level is enormous and may be beyond human ability to construct for the next several decades. While the technology is there, the management and system construction techniques are still primitive. And the rapid pace of technology change will hinder the establishment of a sound technology foundation (the Integrated Enterprise Architecture) for the enterprise and integrated system; components of the system will change during the planning stages - before anything can be installed - and necessitate changing the plans.

There are technology challenges as well. For example, the notion of self-reconfiguration based on system performance is a good idea, but to what extent is it possible and practical?
Obstacles and issues include:

- o Ennui and entropy
- o Sense of completeness or destination
- o Maintaining relevance of metrics as system evolves
- o Specialized technology requirements, such as self-configurable systems
- o Dealing with system complexity

3.5.2 What must be done to maintain level 5

The progression from level 4 to level 5 means that metrics and measurements are used to improve system performance, and that continual change and improvement is part of the routine operation of the enterprise. To maintain operation at level 5, there is nothing really left to "do" to the integrated system. Actions required to maintain level 5, therefore, do not involve changes to the system, but rather changes to the way the system is viewed; maintaining level 5 requires:

- o Vigilance
- o Maintaining holistic view of integrated system, enterprise, and environment
- o Commitment to management of change (i.e., evolution)
- o Recognition that a static sense of completeness or destination means rapid obsolescence

These "actions" are non-material, the results intangible, and the execution dull. In addition, they are very difficult perform over long periods of time. Even the benefits are hard to recognize and may be taken for granted. However, failure to perform these actions is painfully obvious in the deterioration of the system, the loss of business, and finally the failure of the enterprise because it "rested on its laurels" for too long.

3.5.3 What software tools can aid the operation at Level 5

At higher levels of Integration Maturity, it becomes increasingly difficult to identify specific software tools and software requirements because the roles that software plays become

so specialized. The capabilities of software installed at lower levels will obviously need to be expanded and upgraded, but functional requirements for new software will depend on the nature of the enterprise. Many of the software tools used to progress from level 4 to level 5 are also used in the operation of level 5.

Based on the description presented here, there are some general purpose software tools and software requirements that could aid an enterprise operating at level 5. They all fall under the general description of "System reconfiguration support" and can be broken into three categories:

- o System analysis and decision support

- o System modification and upgrade tools

- o Model-based tool and application software

The last item is a software requirement rather than a functional description or the software; it applies to both to the first two items.

Analysis and decision support tools are those that analyze the gathered performance data and provide input to decision support tools. The decision support tools aid the system designers (e.g., members of the IEG) in evaluation system reconfiguration responses to changing conditions. Because the domain is well understood, Expert Systems can effectively be used in this role.

The modification and upgrade tools are the wrenches and screwdrivers used by system implementations to build interfaces, integrate tools and applications, and "make the system work". Beyond the compilers and debuggers used at every level, reverse engineering tools are one of the most important tools in this category. Another important kind of tool is "jack-of-all-trades" software, like a very general purpose, bit-bucket database; this kind of software is used to off-load system functions during upgrade or modification.

Model-based software is a software requirement that recognizes two important characteristics of software system evolution. The first is that system analysts and designers will increasingly rely on models as systems continue to grow in complexity. The second is that the easiest way to modify the performance of a system is to modify its control strategy; by basing the execution of a system on a model and treating the model as a control strategy for the system, the design functions and the operation functions are effectively joined through the model. Changes to the system can be rapidly propagated throughout the system by coordinated releases of new models. (The models should be compatible with - if not the same as - the models used to understand, analyze, and document the enterprise, e.g., process, information, resource models.)

In summary, tools that aid the operation at level 5 include:

- o Integrated system performance analysis and system design decision support tools (e.g., Expert Systems)

- Programming aids to facilitate analysis and integration of software, such as viewers and reverse engineering tools

- General purpose, "jack-of-all-trades" tools to off-load functions during system state transitions

- Applications and tools based on models

4.0 Conclusions

An objective of this chapter was to present a framework for enterprise integration that assumes an evolutionary perspective rather than a static state perspective. The CIM-OSA Reference Architecture and Zachman/Sowa's Information System Architecture are both valuable contributions to enterprise integration efforts, but do not (directly) incorporate the aspect of time in the use of the architecture for system development. Industrial enterprises are messy places that don't tend to fall into the nice, well-defined frameworks defined by the architectures. The Integration Maturity framework presented here attempts to address the "messiness" by abstracting a series of evolutionary stages that describe various characteristics of the enterprise at different stages of growth and provide a basis that enables and fosters evolution of the enterprise and enterprise systems.

The primary purpose of this chapter was to identify software tools and software requirements for manufacturing enterprises. While much of the chapter is not devoted to software (and almost none to manufacturing, per se), the intent of presenting the Integration Maturity framework is to illustrate opportunities for software development with respect to enterprise integration. Because of the prevalence of manufacturing automation, manufacturing enterprises are in a particularly receptive position to benefit from enterprise integration.

The software tools that were identified across the levels exhibited an interesting characteristic. At the lower level, the tools that could best aid the evolution of the integrated system were very general-purpose tools that were used to analyze, understand and document the technology and the enterprise. At the middle layers, the tools were more specialized and tailored toward integration and information management. At the higher levels, the tools again became more general purpose, though this time with a different focus. The general purpose tools at higher levels were not intended for understanding the system, but working with and as part of the system; rather than being objective tools, they are subjective tools that are part of the system (not unlike the roles of maintenance personnel in a building).

An important point to note about the work presented here is enterprise systems do not and cannot evolve independently of the evolution of the enterprise as a whole. The improvement recommended here must take place within larger enterprise improvement plans. Improving the level of integration in an otherwise poorly designed enterprise will not dramatically improve the enterprise, will be more difficult, and could lead to undesirable consequences. Like professional sports, playing in the big leagues before doing time in the minors can lead to career-ending results. Address the larger problems first.

The Integration Maturity framework owes a debt to the work of Watts Humphrey and Philip Crosby. The evolution perspective used their work provided the genesis for this analysis of integrated system evolution.

Bibliography

[1] Crosby, Philip B. Quality is Free - The Art of Making Quality Certain, McGraw-Hill, New York, 1979.

[2] ESPRIT Consortium AMICE (Eds.) Open System Architecture for CIM, ESPRIT Research Reports, Project 688, AMICE, Volume 1. Springer-Verlag, Berlin, 1989. ISBN 3-540-52058-9.

[3] Humphrey, Watts S. Managing the Software Process, Addison-Wesley, Reading MA, 1989. Software Engineering Institute Series in Software Engineering. ISBN 0-201-18095-2.

[4] Zachman, J.F., and Sowa, J.A. "Extending and formalizing the framework for information systems architecture", IBM Systems Journal, Vol 31., No. 3, 1992.2

Chapter 5
Manufacturing Enterprise Design and Data Management

Marc Halpern
Program Manager, MCAE, CAD/CAM
D.H. Brown Associates
222 Grace Church Street
Port Chester, NY 10573
914/937-4302

Seth Hunter
Senior Engineer
Miles Diagnostic Laboratories
511 Benedict Avenue
Tarrytown, New York 10591
914/631-8000

Abstract

Mechanical Computer Aided Design has undergone tremendous upheavals in the past decade due to technical breakthroughs such as solid modeling, simulation software, and product data management. Companies worldwide have adopted these technologies, buttressed by explosive growth in the performance of desktop hardware, as a means to shorten their product development cycles. Solid modeling offers the potential of unambiguous product definition enabling the elimination of drawings. Simulation software promises proactive detection of performance problems early enough in the design cycle to make changes inexpensively. Product Data Management offers control of voluminous computer generated design data as well as a segway to optimal management of the design process.

While these technologies have offered significant advantages, companies soon discovered that they also pose significant challenges. Design processes must change considerably to capitalize on the new generation of software. Also, significant technical challenges must be answered. This work briefly discusses current software technology and practices for mechanical design. It also identifies the technical barriers yet to be overcome. Conclusions are based on in-depth reviews of leading edge technology and detailed surveys of user requirements and practices conducted by D. H. Brown Associates.

Key Words

Constraints; Configuration Management; Document Control; Engineering Process; Enterprise Solution; Features; Idealization Errors; Product Data Management; Simulation Software; Solid Modeling; Virtual Prototyping; Workgroup Solution

1. Introduction

Intense global competition has caused major manufacturing enterprises worldwide to seek new ways for cutting costs and reducing their product development cycle to remain competitive. Major objectives of these strategies include definition of a common design database, early detection of design problems through virtual prototyping, and concurrent engineering. Key technologies to accomplish these goals include solid modeling, simulation software, and Product Data Management (PDM).

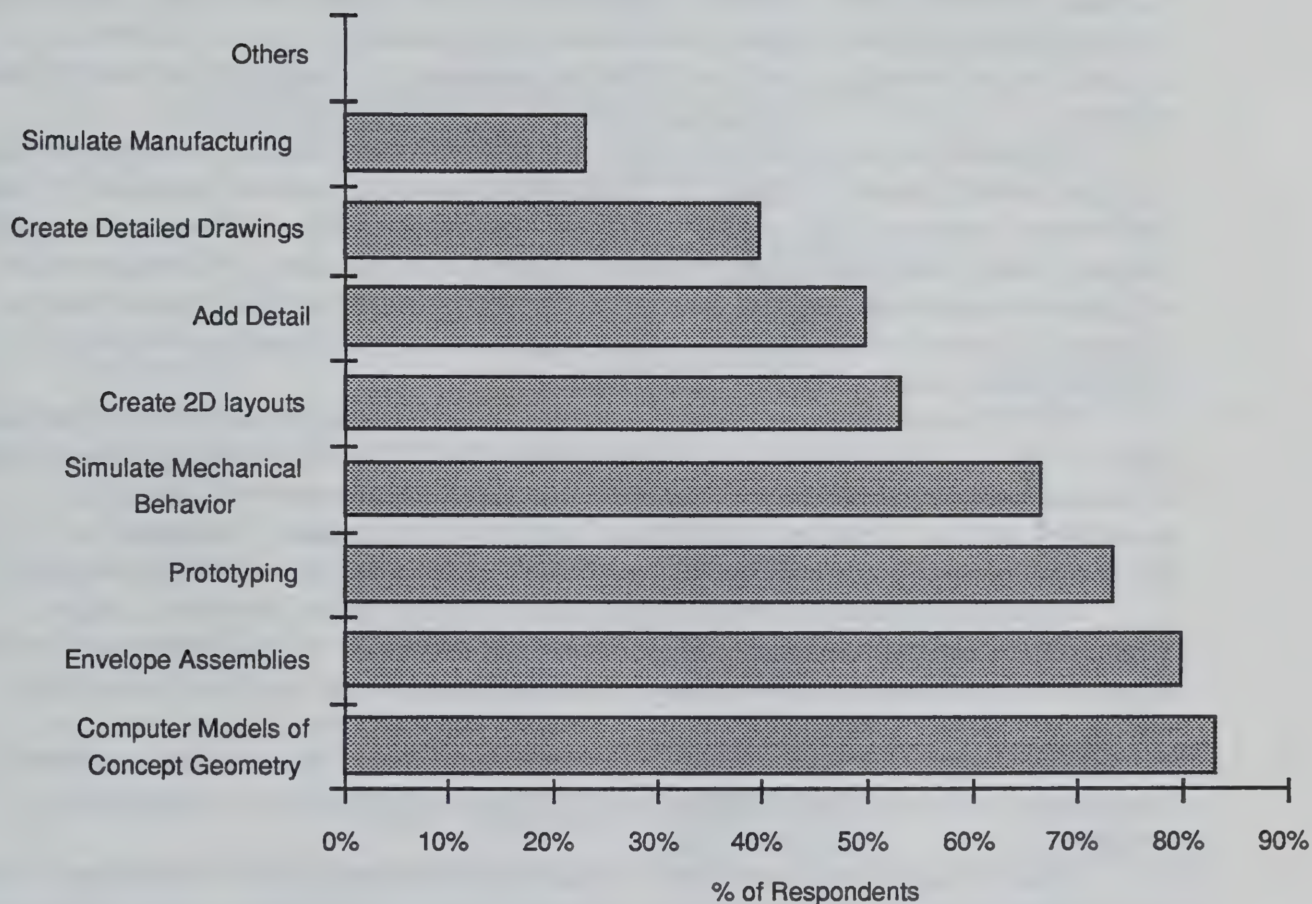
The worldwide manufacturing base has been attracted to solid modeling software because the solid model potentially unambiguous design definition. Solid modeling eliminates the endemic problem of drawing mis-interpretation that leads to manufacturing problems. Unambiguous design definition also suggests that solid modeling can potentially drive downstream applications such as simulation, manufacturing operations, generation of documentation, and marketing functions. Indeed, leading corporations talk of "virtual prototyping" through solid modeling as a new approach to design that can dramatically reduce time and expense for new product development. Virtual prototyping involves the use of solid models to simulate manufacturing operations, product assembly, and simulation of product performance in its operating environment before a company commits any resources to materials or manufacturing equipment. Ideally, the early investment in virtual prototyping saves substantial time and resources in later product development.

PDM plays a critical role in the drive for concurrent engineering. Functionality such as document control, configuration management, engineering process control, and systems management enables the enterprise to effectively share information and managed product definition data from design conceptualization through manufacturing including materials requirement planning.

However, significant technology gaps must be bridged before companies achieve the goals mentioned. Solid modeling and simulation technology has been unstable and continues to evolve. Today's capabilities do not fully address the requirements of industry. Also, virtual prototyping involves substantial changes in corporate culture and design processes. These changes in process will also impact evolving PDM technology. This publication identifies those gaps as they relate to design conceptualization, simulation of product performance, and management of electronic design data.

2. Design State-of-the-Art

Between 1991 and 1992, DHBA surveyed 45 large corporations in the U. S. and Europe across diverse industries including aerospace, automotive, electronic components, white goods, consumer electronics, medical equipment, and research laboratories [23]. Surveys were conducted through telephone interviews, on-site visits, and written surveys. These large corporations were selected based on their commitment to adopting leading design technology and their diversity. Figure 1 summarizes the most common activities for design conceptualization.


Others
Simulate Manufacturing
Create Detailed Drawings
Add Detail
Create 2D layouts
Simulate Mechanical Behavior
Prototyping
Envelope Assemblies
Computer Models of Concept Geometry
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
% of Respondents

Creation of design geometry represents the single most time intensive activity of conceptual design. Leading edge users in conceptual design demand fast and flexible editing first and foremost. Nothing yet matches pencil and paper. As designs crystallize and design rules become complex, the decision arises as to 2D or 3D modeling. The nature of the design, and often the personnel involved, dictate which mode is selected. If drawings are adequate for communication, if work does not involve non-technical personnel, and if the company manufactures directly from drawings, designers will probably find the 2D mode most efficient. When the design becomes complex and difficult to conceptualize on paper, or when the designer has to communicate with non-technical personnel, 3D design will probably be most efficient.

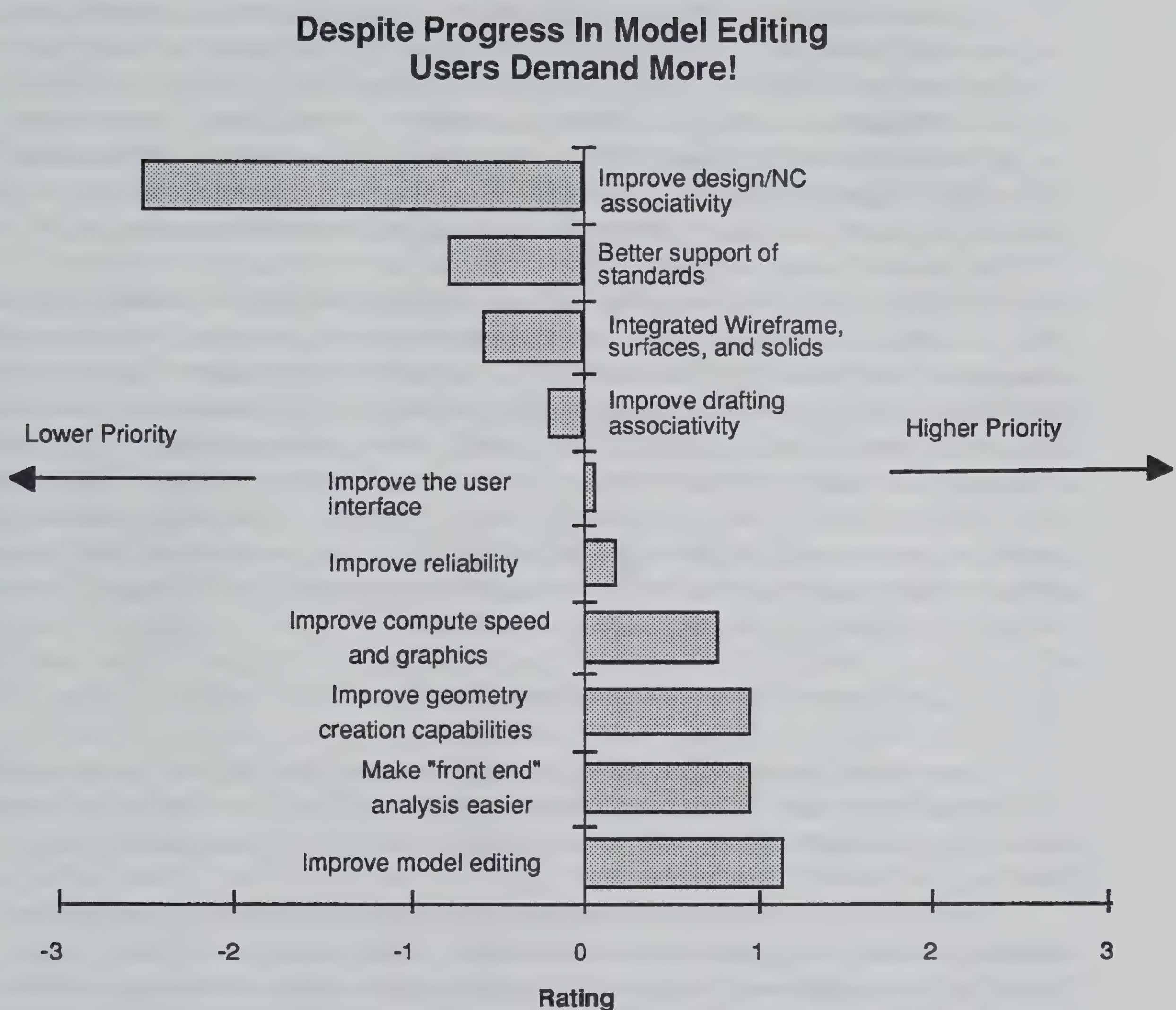
Engineers and designers work with a mix of wireframe modeling, surface modeling, and solid modeling to create geometry. The preferred modeling method depends on the design. For example, engineers might prefer wireframe modeling for structural frames. Automotive designers prefer surface modeling to develop Figure 1 aerodynamically efficient car bodies with complex sculptured surfaces. Solid modeling works well for a broad class of objects. However, it works best for relatively simple shapes. In many circumstances where solid modeling works, solids offer the quickest editing.

Simulation of mechanical behavior refers to use of techniques such as the finite element method, boundary element method, volume element method, and the finite difference method to simulate mechanical behavior of a design in its operating environment. Today, this technology requires a relatively high level of expertise to use reliably. No simulation technology has yet been introduced that can be used reliably and expediently by designers.

Envelope assemblies refer to conceptualization of the space layout for a design. Engineers and designers will use solid modeling to conceptualize a three dimensional product layout though volume envelopes to insure that components do not interfere. For example, during early stages of automobile design, a designer may use a simple shape such as a hexahedron to represent a carburetor. As the design progresses, engineers add details and components to the carburetor such that the final design resides within the original design envelope.

Figure 2 summarizes technology that users prioritize for further development. Priority for improvements for each categorywere compared to an overall average across all categories. All categories to the right of the vertical axis have higher than average priority. All other categories have lower than average priority. Improvements in model editing, front end analysis capabilities, geometry creation, and software performance were rated highest.

Although substantial progress has been made in model editing, DHBA research determined that electronic capture of concepts and editing have not have not matched user requirements. the majority of designers and engineers do early conceptual design with pencil sketching because of the instantaneous response to changing ideas. They use CAD tools only as design concepts stabilize. Companies report that they would prefer designers to use computer more often for conceptual design. As an advantage, computer based conceptual design would eliminate duplication of geometry creation from paper to computer when design


Despite Progress In Model Editing
Users Demand More!
Improve design/NC associativity
Better support of standards
Integrated Wireframe, surfaces, and solids
Improve drafting associativity
Lower Priority
Higher Priority
Improve the user interface
Improve reliability
Improve compute speed and graphics
Improve geometry creation capabilities
Make "front end" analysis easier
Improve model editing
-3
-2
-1
0
1
2
3
Rating

concepts have stabilized. The early use of any commercial software depends on its ease of editing. Opportunities exist for substantial technical innovation since no software product today offers a complete solution that spans conceptual design requirements and detailed design requirements.

As a separate issue, solid modeling relates to unified databases and unambiguous product definitions. Users accord high priority to better geometry creation capabilities for complete capture of designs as solid models for unambiguous documents of record. To date, none of the modeling technology available has the capacity to capture all design details. Some companies occasionally compromise on their designs because modeling software could not capture their original design as a solid model. Since they want the solid to be the design document of record, they decide to modify the design instead of creating proper solid as long as they do not compromise design function.

Making "front end" analysis easier has the greatest potential for technical innovation. "Front end" analysis refers to simulation of mechanical performance in a product's operating environment. Companies hope to detect mechanical performance problems as early as possible before they commit substantial resources to any particular design. Unfortunately, since the tools require significant time to prepare the analysis and significant expertise to do it correctly, simulation tools seldom get used this way. Instead, companies use simulation software for design verification at the end of the design process. Specialists in simulation technology typically take responsibility. For effective use early in product development, the simulation software must be made easy and reliable enough for designers. Sections 5 and 6 cover simulation technology in greater detail.

2.1 Constraints and Feature Based Modeling

There have been significant breakthroughs in mechanical design software technology over the past eight years. Before that, poor editing plagued CAD, particularly solid modeling. The most significant breakthroughs include constraint based design and feature based modeling.

The combination of constraint based design and use of features opened the door to solid modeling for many corporations worldwide. Before the introduction of constraints and features, editing solids was a major barrier to productivity. Corporations began to embrace solids after Parametric Technology Corporation [24], introduced commercially available constraints and feature based design in 1987. Today most leading commercial CAD vendors including Autodesk [11], Computervision [20], Dassault Systems/IBM [10], EDS/Unigraphics [12], Hewlett Packard, Intergraph [15], Matra Datavision [22], Parametric Technology Corporation, and SDRC [17] offer this new technology.

2.1.1 Constraint Based Design

Constraint based modeling enables users to "program" their designs. This programming captures relationships defined by constraints among design variables and geometric entities. For example, parallelism, perpendicularity, or tangencies between curves

and lines represent geometric constraints. The length and width of a rectangle serve as examples of design variables. A user can create the rectangle as geometry and then relationships among length and width as design variables that drive changes in the geometry. One can establish relationships such as length = 2 X width. Therefore, anytime a designer changes the rectangle's width, the length automatically updates the length properly.

Designers typically want to maintain the design intent that constraints capture throughout the design process. In earlier CAD systems, a change in the length, shape, or position of a curve would typically destroy such established geometric relationships. The designer would have to laboriously recreate much of the existing design manually to re-establish those relationships. Constraint based modeling preserves those relationships. An entire design can automatically update when one entity changes.

Parametric and variational methods represent two approaches to manage constraints. Parametric constraint management refers to the sequential resolution of constraints. Constraints can be resolved in an order dependent way that reflects "cause and effect." Changes downstream can never modify variables defined earlier. Variational constraint management refers to the elimination of such causality and order dependence. Variational systems solve for conditions and variables simultaneously. Changes to design variables downstream can modify variables defined earlier. Differences in implementation have been reviewed in earlier work [25].

Variational systems also allow more flexible constraint strategies than parametric systems. Sequential solutions in parametric systems limit the solutions to a subset of all the acceptable possibilities. Practically, this subset consists of only those that can be sequentially constructed on a drawing board using a ruler and compass [8]. Variational approaches allow a broader range of strategies that are relevant in design and mechanism analysis.

Variational systems require the iterative numerical solution of nonlinear systems of simultaneous equations. Today, variational approaches have been streamlined to improve computational efficiently. However, numerical solution techniques still carry the specter of accuracy problems. Parametric approaches have no such risks. Also, parametric systems are easier to diagnose if a user committed an error applying constraints. Since these systems reflect "cause and effect," users more easily find where an error was made.

Today, improvements in parametric systems facilitate changes in constraint strategies. This eliminates one of the advantages of variational systems. Parametric systems such as Hewlett Packard's ME-10P [19] allow users to change constraint strategies. The software automatically searches for the new sequence of "ruler compass" constructions that supports the new constraints. This approach prevents users from being locked into any constraint scheme over the design cycle. Therefore, the early concerns that parametric approaches lock users into a given constraint scheme have been alleviated. All of the systems reviewed allow flexibility of change.

Differences in parametric and variational modeling affect users primarily in terms of the allowable constraint strategies, in terms of the diagnostic feedback to resolve constraint application problems, and in the accuracy of the constraint resolution procedures. The quality of implementation also impacts the product. A strong implementation of parametric modeling is more useful than a poor implementation of the variational approach.

2.1.2 Feature-Based Modeling

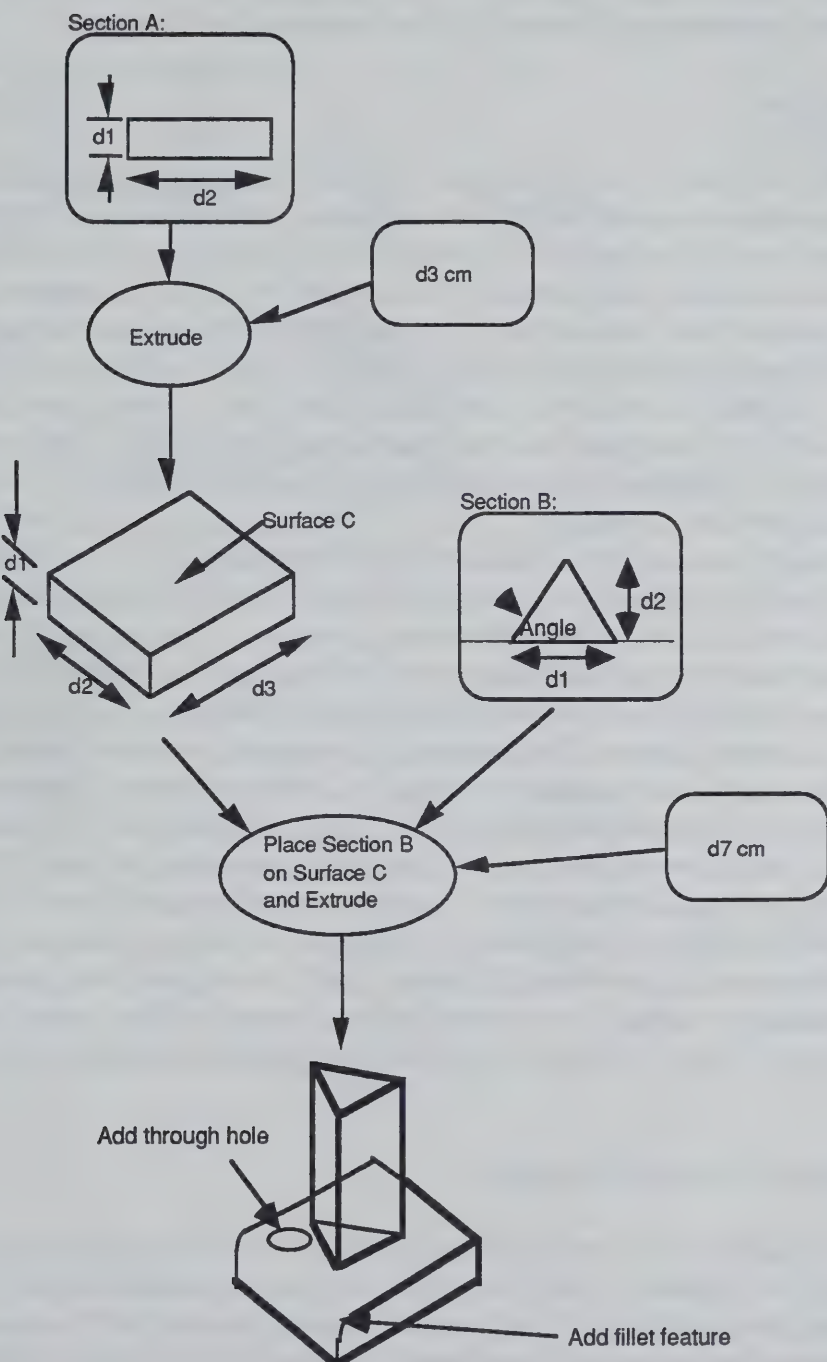
Feature based modeling represents the encapsulation of geometric and non-geometric design information into a single entity called a feature. Non-geometric information can include construction operations and constraints. Features can also include information for downstream applications such as NC or engineering analysis to reliably expedite generation of machine tool paths or finite element meshes. A robust feature based modeling system encapsulates the right information for features to dynamically update a model and maintain design intent. Users benefit from feature based modeling through improved ease of use, better model editing, and greater reliability that software retains design intent when modifications are made.

Designers and engineers think in terms of features such as fillets, slots, holes, chamfers and ribs, etc., when creating designs. They do not think in terms of the more fundamental and abstract geometric forms or operations required to create those features. When designers think of fillets, they envision rounded edges having specific radii. They do not instinctively think of the geometry creation or modification activities required to create that fillet. The fillet feature transparently imbeds all of the operations required to create the correct geometry. Designers only have to specify the location of the fillet and its radius. Feature based design frees the designer to think in terms of design requirements and not in terms of geometry construction.

2.1.3 Modeling Workflow with Constraints and Features

Figure 3 illustrates a typical workflow combines parametric/variational constraint based modeling with features. First, the designer creates a cross section of rectangular Section A shown at the top and then applies constraints. Constraints include definition of parameters d1 and d2 as shown to define the lengths of the two sides. When the final solid has been created, a simple change in d1 or d2 will update the entire solid. Additional constraints to define the rectangle can include definition of parallel sides, perpendicular sides, vertical sides or horizontal sides. Any combination of constraints that defines the rectangular shape is acceptable.

Next, the designer extrudes the cross section into a solid. Section B, a triangular cross section, might be sketched and constrained in a fashion similar to Section A. Ideally, the designer should be able to sketch Section B directly on the top surface of solid block generated from Section A as in Pro/Engineer. Well designed software allows users to reverence existing geometry to position the triangular cross section. After extruding the triangular profile into a new solid feature, the designer references existing geometry to add the fillet and through hole. Since a feature based modeler understands the concept of a hole and a fillet, the designer must only provide the position of the feature and critical information that defines it. When existing geometry changes, features dynamically adjust, maintaining design intent, since features understand their relationships with pre-existing geometry.


Section A:
d1
d2
d3 cm
Extrude
Surface C
d1
d2
d3
Section B:
Angle
d2
d1
Place Section B
on Surface C
and Extrude
d7 cm
Add through hole
Add fillet feature

Assemblies can also be defined with constraints. Products from vendors such as Parametric Technology Corporation [24] and SDRC [21] allow definition of relationships between components through both sketches or by applying constraints directly to objects. Figure 4 below illustrates typical assembly constraints applied to objects.

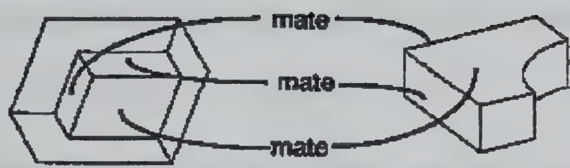
3. Design Gap

Recent survey work by D. H. Brown Associates on actual use of solid modeling in major North American corporations [13, 14] suggests significant technology gaps. These technology gaps relate to challenges of communicating design intent to all members of a product development team. Also, ability to use the solid model as the standard across the enterprise for diverse functions such as design, engineering, manufacturing, and materials purchasing poses broader challenges. Users must know how a constraint-based feature based solid model is constructed in order to work with it effectively. Although this problem can be alleviated by enforcing strict guidelines for constructing solid models, such guidelines will not solve the problem. Technological innovation that address this issue will accelerate the productivity achievable with solids.

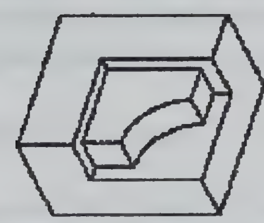
For example, the ability to capture design methodology and design intent within a design data exchange could capture a design knowledge base. Designers and engineers embed knowledge into the data exchange standard through the use of features and constraints as a matter of course in developing the initial design, without additional effort. All coordinated design and manufacturing functions, across departmental boundaries, may access a complete definition of the current version. The database, incorporating constraint based and feature based capabilities, serves as a primary enabling technology to support concurrent engineering in practice. Further, companies can easily retain design expertise after a valued designer or engineer has shifted to new projects, or even left the organization. By more fully incorporating design expertise, the database encourages the reuse and refinement of established design practices. Aside from directly contributing to dramatic cost savings and quality improvement, the approach fundamentally provides the means to fully leverage evolving technology, and to employ that technology directly to the organizational constraints that burden existing design and manufacturing efforts. The following technology gaps must be addressed to achieve these goals.

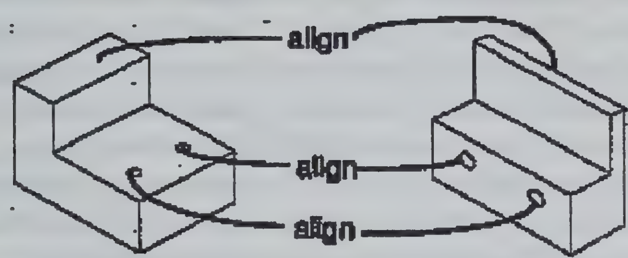
3.1 Sharing Features and Constraints

Most companies do not use just one CAD system. They need to share information across multiple CAD systems to fully serve the enterprise. Standards such as ISO STEP succeed at transferring surfaces of a solid model across disparate CAD systems. However, ISO STEP does not support features and constraints. The unfulfilled technical requirements of sharing constraints and features across different CAD products suggests that ISO STEP will not address this technology gap for years to come. Fundamental technology advances must be made first.

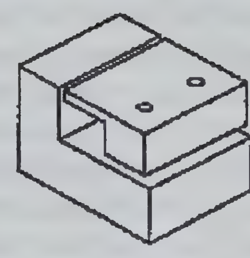


Yields:





Yields:



This gap poses a long term serious problem for industry since companies have been rapidly building product definition databases with features and constraint information. Constraints and features contain most of the design intent associated with a design. Also, feature definitions are evolving to incorporate more "non geometric information" to support manufacturing, engineering analysis, real time cost estimating, producibility analysis, etc.

3.2 Leveraging Archived Solid Models

Since many new products and parts evolve from existing designs, geometric shapes should permit more than archival storage of the final design representation, which then serve analysis and production requirements in a separate phase. Rather, electronic representations should allow modification and feature editing. Unfortunately, no solid model representation in use today includes the required capability. All editable representations used by commercial and research CAD systems are proprietary, and involve data definitions at too low a level of abstraction. Also, current representations for solid models require large data files. For example, the editable representation of the crankshaft of the DARPA diesel engine is over 6 Mega bytes in Pro/Engineer [7]. A high-level representation of the design could easily be devised using less than 1/100th of this space. More compact representations could also facilitate distributed and geographically dispersed design and manufacture, an increasingly more important consideration with the emergence of virtual factories and the national information infrastructure.

To accomplish these goals, an information substrate is required that captures both conceptual design rationale as well as design structure needed to redesign product families and instantiate product variants. Information should specify how to derive a product design instead of simply defining the design specification itself. The information must include how to derive a geometric shape, the structure of the shape, information about the functional intent, and engineering requirements. Key aspects of the information substrate include:

- The ability to archive information in a manner that does not commit to specific modeling technologies.

- The ability to express generic and conceptual design in a manner that is targeted to the requirements of broad application areas instead of current technological ability of a particular software system.

- The ability to solve and enforce constraints on the instantiation of design, including 3D constraints.

These aspects must be realized using general approaches that apply broadly across engineering and product design.

3.3 Open CAD Architecture-Technical Challenges

Significant technical problems must be solved before an open CAD architecture can be developed. The most critical of these include 1) the persistent name problem, 2) the root identification and degeneracy problem, and 3) three dimensional constraint solving.

3.3.1 The Persistent Name Problem

Modifying design operations such as blending an edge requires the designation of an element of the shape instance to be blended. When design parameters change and a new instance is automatically compiled, the edge is no longer there. Worse, the edge need not explicitly correspond to a design gesture, for instance when arising from a subdivision of the intersection of two features by a third. The problem is illustrated by Figure 5. The example is from Pro/Engineer, the industry leader in feature-based, generative design. On the left, a block has been designed with a chamfered edge and a cut across the top face. A hole is cut by revolving a rectangle drawn in the datum plane parallel to the front face and dimensioned to be at distance 200. The back edge of the hole is rounded. On the right, the placement of the hole is varied slightly, by decreasing the distance from the front face to 180 degrees. On regeneration with the changed dimension, Pro/Engineer decides erroneously to round the front top edge of the hole .

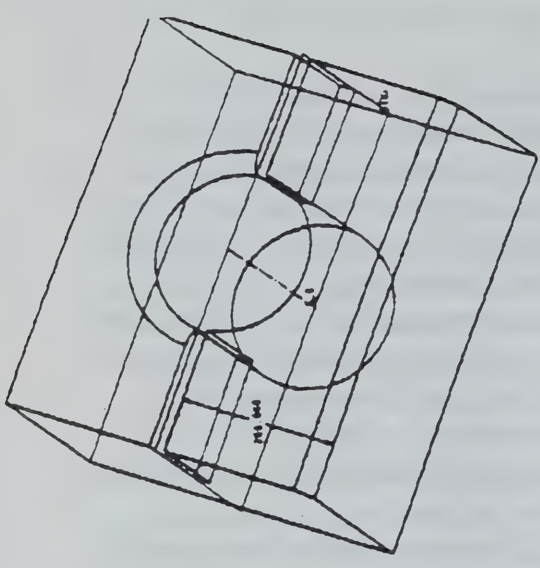
A generic naming scheme must be devised that is capable of identifying the geometric instance elements on regeneration from a changed design.

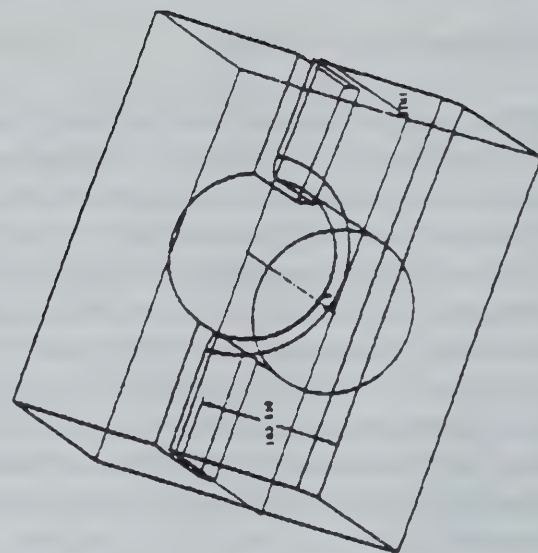
3.3.2 Root Identification and Degeneracy Problems

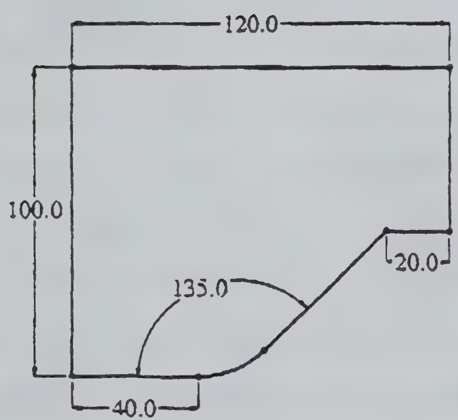
It is well known that design constraints require the solution of systems of nonlinear equations, possibly exponentially generating many solutions. Moreover, enforcing even minimal design validity automatically becomes intractable and cannot be done efficiently without help from the user. We have explicit rules that select the intended design solution, and techniques for visually interacting with the constraint solver when the solution is not the desired one. Our interaction techniques are intuitive and do not burden the user with the detailed mathematics of the constraint solver. Simple solver errors are illustrated in Figures 6 and 7. The examples are from DCM [8], a successful commercial constraint solving engine.

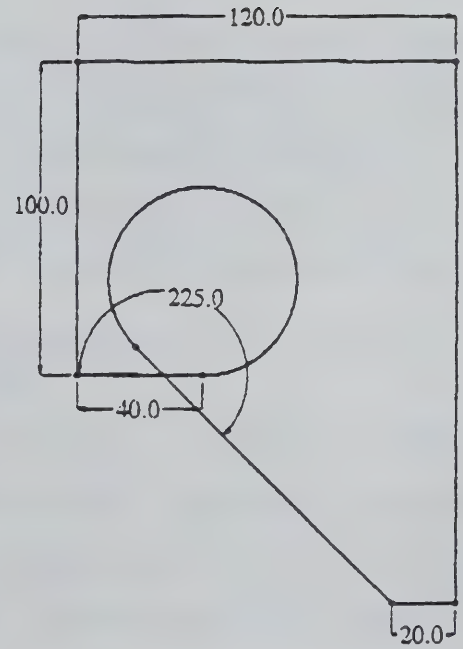
In Figure 6, two edges are rounded by an arc. On the left, the original drawing requires that the center of the rounding arc be on the left of the counter-clockwise oriented adjacent segments. When the angle between the segments is altered, this topological rule for placing the rounding arc should be suspended in the construction of the new solution. As the right picture demonstrates, DCM is not able to derive the correct solution.

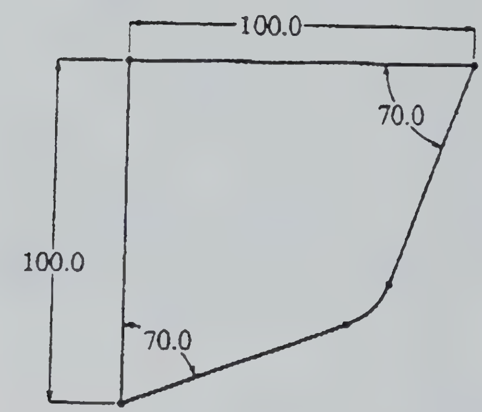
Figure 7, the rounding arc must have zero length in certain situations. The critical angle of the right instance is such a case, but DCM insists on an arc of non-zero length and puts in a full circle.

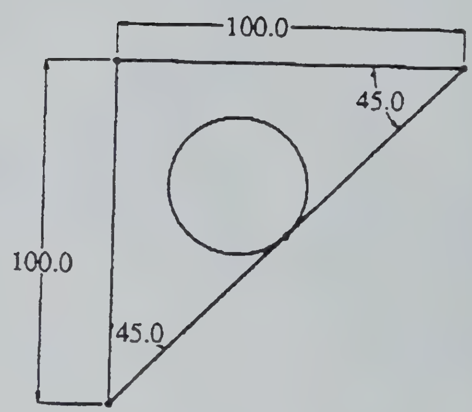





120.0
100.0
135.0
20.0
40.0


120.0
100.0
225.0
40.0
20.0


100.0
70.0
100.0
70.0


100.0
45.0
100.0
45.0

3.3.3 Three-Dimensional Constraint Solving

While the mathematics of two-dimensional constraint solving is well established, practical algorithms for variational, three-dimensional constraint solving are in their infancy [4]. Yet this area has significant pay-offs that impact design and assembly in virtually every aspect.

3D constraint solutions enable conceptual design. In CAD, abstractions of features that have functional significance often are simple geometric quantities such as axes of revolution or lines of compliance. These abstract features can be specified in their spatial interrelationship in parts and in assemblies, and preliminary analyses can be carried out on them. The language of specification is naturally the language of 3D geometric constraints.

Tolerances on the interrelationship of the functional elements critically affect the cost of manufacture and the performance of an assembly. If we compile the nonlinear system that expresses the 3D constraints and do a sensitivity analysis on it, the results establish how system parameters affect positional accuracy, yielding insights into tolerance requirements.

3.2 Mechanical Simulation Practice

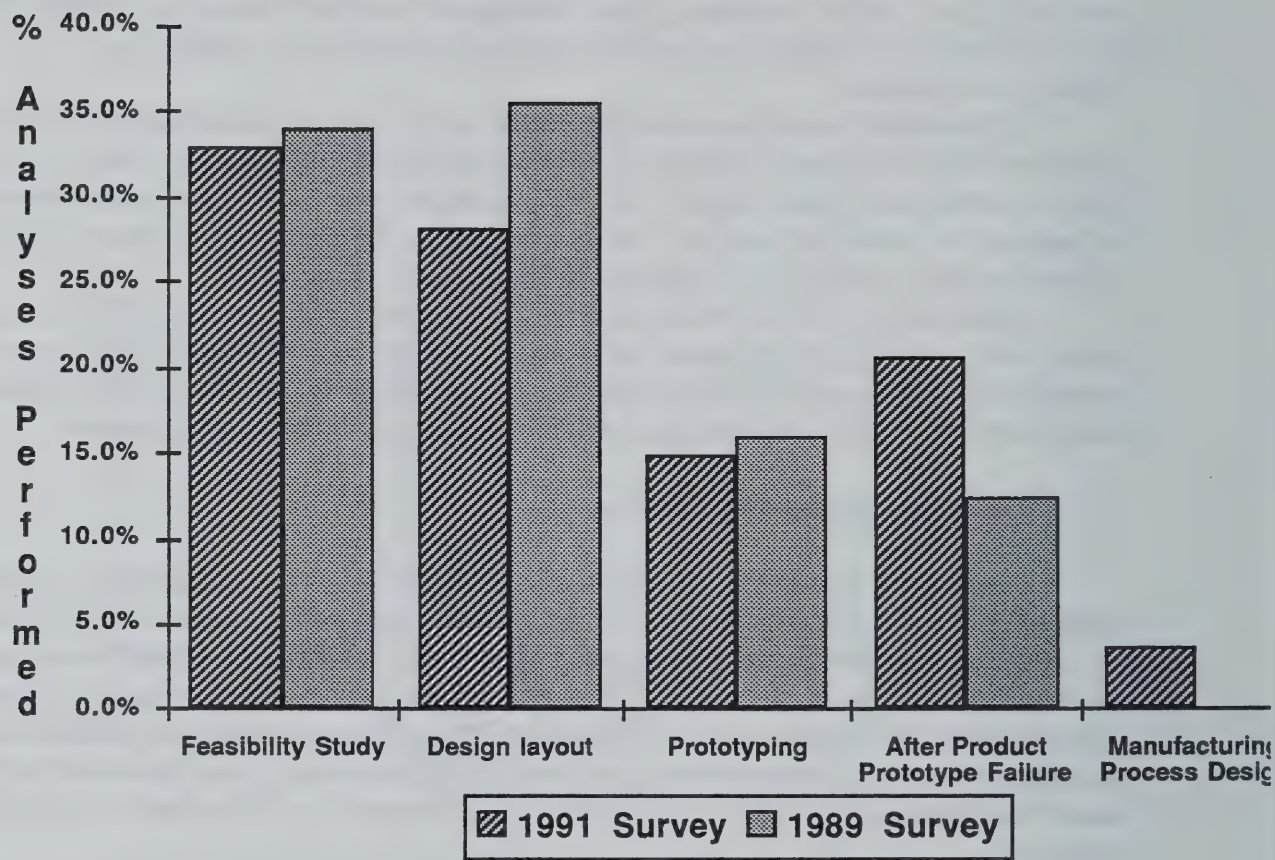
Commercial simulation tools, critical components of virtual prototyping, evolved rapidly for more than twenty-five years, and allowed engineers to analyze problems that could not be solved in before digital computers. Among these technologies, finite element technology has the most widespread use. During 1991, D. H. Brown Associates conducted a survey to understand the use of finite element technology [26]. The survey addressed customer opinions of leading commercial products and also identifies key technology gaps that inhibit effective use of simulation software early in the design cycle. More than 180 companies participated in the survey.

Figure 8 from a survey of finite element practice summarizes the most typical analysis tasks. It compares results from data collected in 1989.

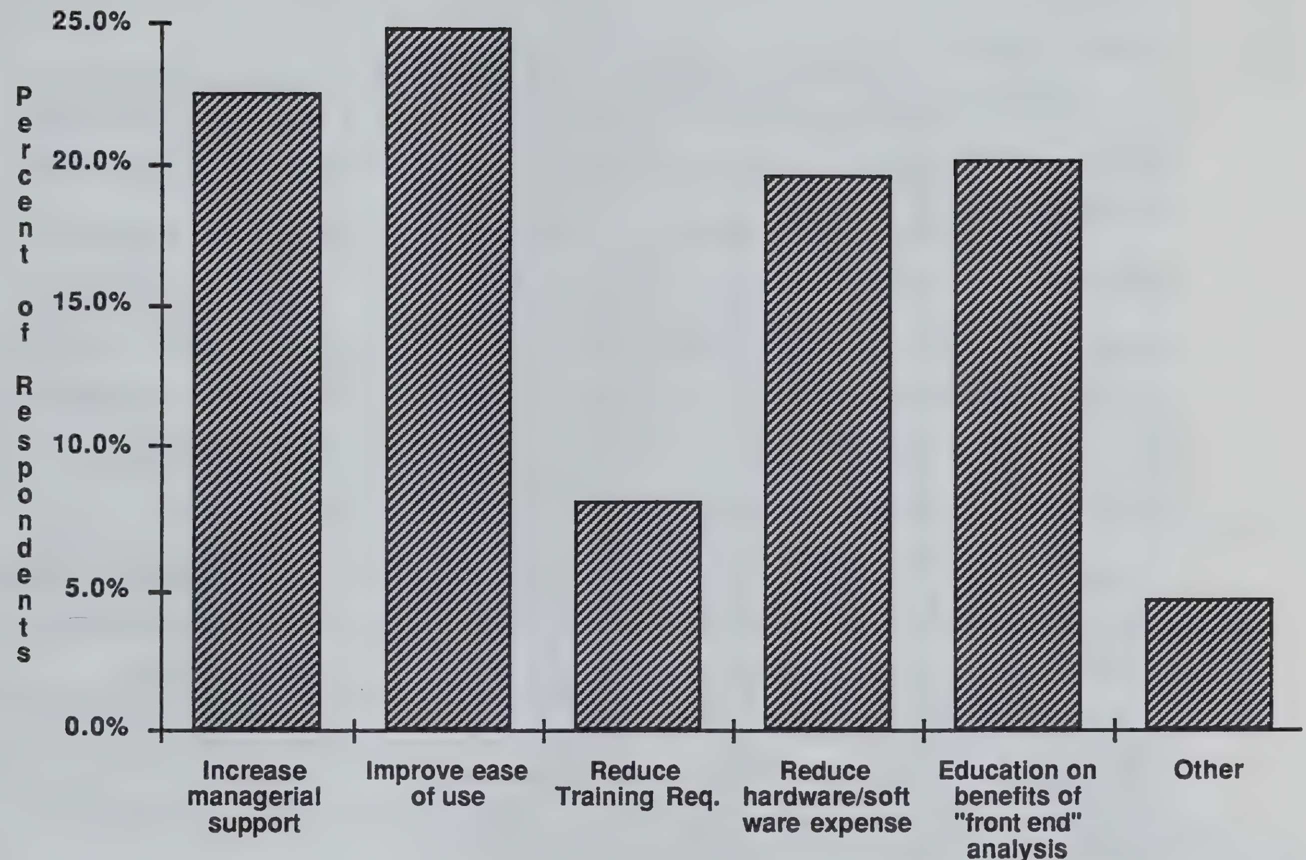
Figure 8 indicates no increase in front end analysis. This relates to technology gaps discussed later. The number of analyses performed after prototype failure has even increased. On a positive note, it may indicate that companies have been using computer based simulation tools more frequently to analyze reasons for design failure. However, another interpretation suggests that the increase in analysis after prototype failure reflects the failure of analysis before prototype testing to predict design problems. The technology gaps identified assume the latter interpretation.

Figure 9 summarizes survey responses to a question addressing requirements to improve effective analysis use.

The greatest challenge relates to software ease of use. Today, simulation software requires a high level of expertise to use effectively. Since most companies have few qualified experts, the available specialists cannot handle all of the needed simulation work. Therefore, simulation technology must be advanced to the point that a broader group of users, including designers, can use the tools. Only then, will simulation technology will achieve its potential for design process improvement.



% Analyses Performed
40.0%
35.0%
30.0%
25.0%
20.0%
15.0%
10.0%
5.0%
0.0%
Feasibility Study
Design layout
Prototyping
After Product Prototype Failure
Manufacturing Process Desig
1991 Survey
1989 Survey



25.0%
20.0%
15.0%
10.0%
5.0%
0.0%
Percent of Respondents
Increase managerial support
Improve ease of use
Reduce Training Req.
Reduce hardware/soft ware expense
Education on benefits of "front end" analysis
Other

In contrast, recent advances in CAD tools such as solid modeling, supported by constraints and features has had a more immediate impact on design practices because they address a broader class of users. Acceleration of the design process through solids exacerbates the challenges of using simulation software. For example, while solid modeling quickly helped companies detect interference problems in mechanical assemblies early in design, management often ignores results of mechanical performance simulations since results arrive several design iterations too late.

Simulation plays the greatest role for design verification. After designs have been completed, management may request a simulation as a check on mechanical performance. However, if simulation results do not match test results, the management will discard the simulation study [3].

Users were also asked where developers should focus their efforts. Figure 10 summarizes the feedback.

Users identified mesh generation as the highest priority for technology development. Mesh generation refers to preparation of a finite element simulation model which represents the most time consuming aspect of performing simulation. Model preparation accounts for the most significant challenge to using simulation software early in the design cycle. Respondents also emphasized the need for improved productivity and better integration of analysis with design products. Such enhancements expedite use of analysis early in design.

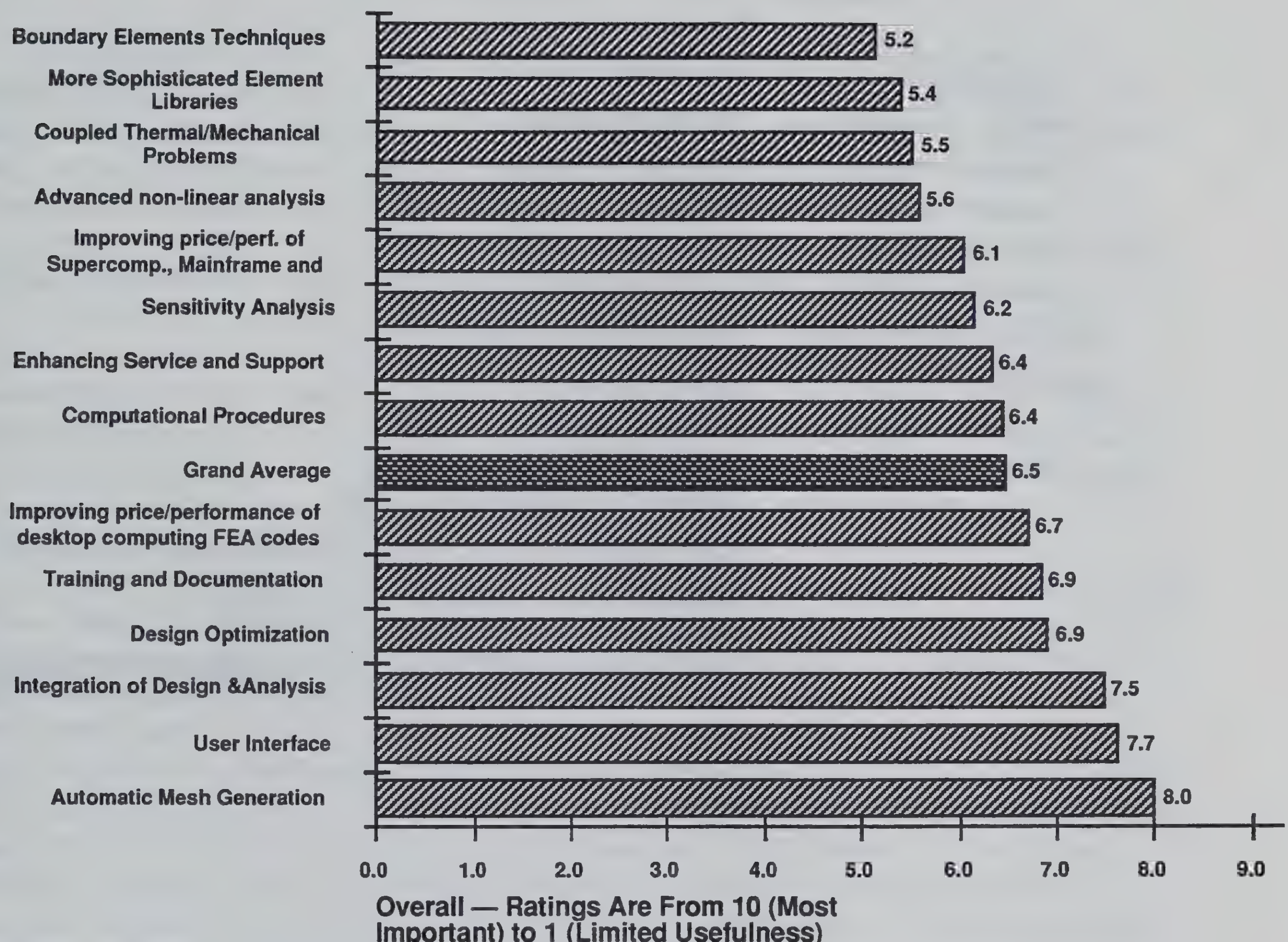
Users also expressed strong interest in design optimization software. Automated mechanical design optimization tools have become commercially viable. Mechanical design optimization software will be the most active research and development focus for the next decade.

Simulation software must address a variety of physical behavior such as mechanical deformation, fluid flow, heat transfer, and electromagnetic behavior to be comprehensive. Analysis experts were asked which physical behavior do they want to use that they have not used before. This question reflects the physical behavior mostcommonly encountered yet infrequently simulated. Figure 11 summarizes the results.

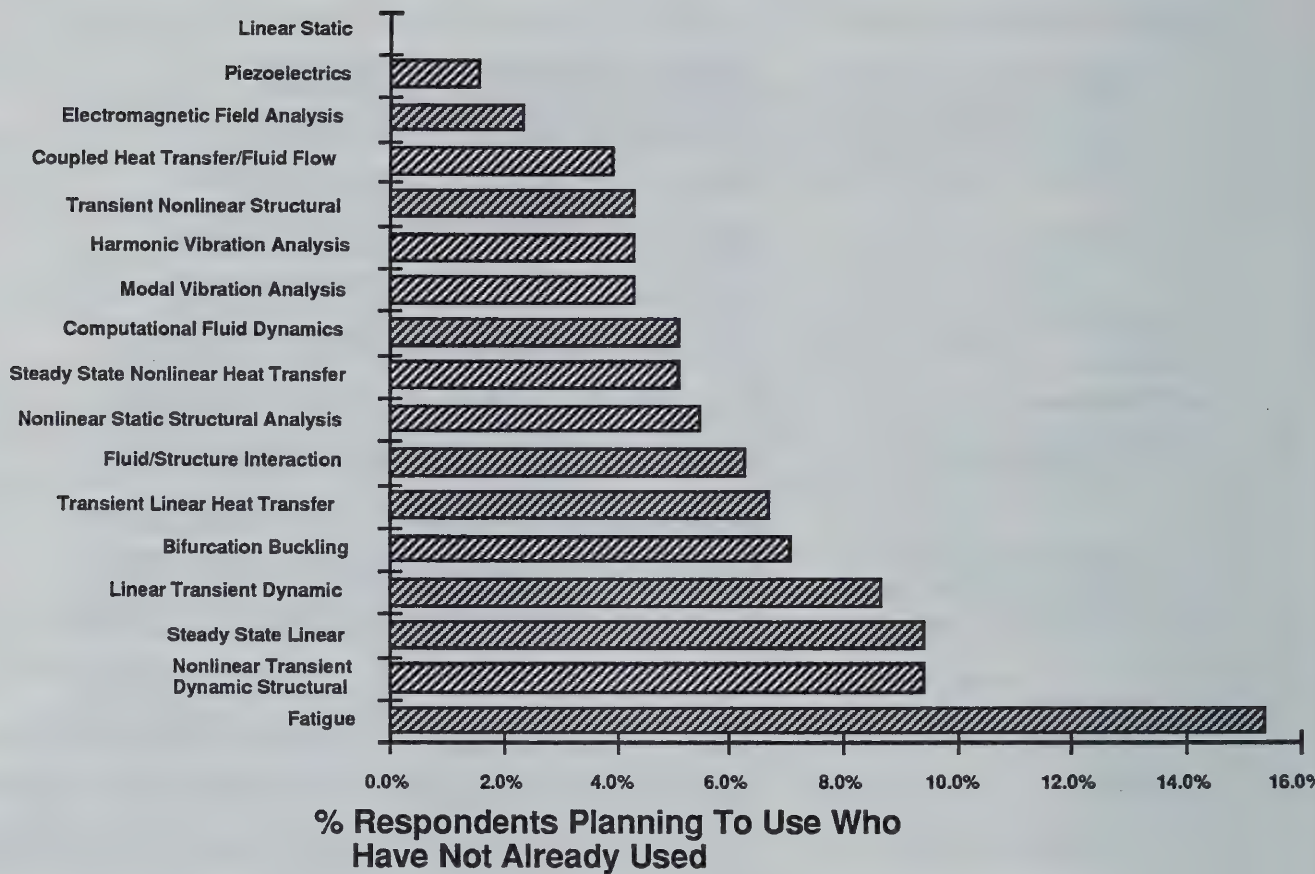
Respondents expressed the greatest interest in fatigue. Fatigue, which addresses material degradation due to cyclic loadings, is one of principal causes of product failure across all industries. DHBA reviews of analysis software indicates that little commercial technology exists that supports fatigue adequately.

3.3 Simulation Technology

Developments in finite element technology during the 1980's have been changing the complexion of simulation in the 1990's. The innovative technical leaders have begun to promote recent breakthroughs in finite element technology that fully capitalize on exploding hardware performance. Also, mainstream analysis vendors as well as startup companies have invested heavily in development to roll out new solution capabilities involving h and p adaptive analysis, hierarchic shell analysis, and comprehensive capabilities to solve problems involving multiple interacting physical processes. The industry has also been moving to tighter integration between CAD and simulation technology. The most progress has been made with



Boundary Elements Techniques
5.2
More Sophisticated Element Libraries
5.4
Coupled Thermal/Mechanical Problems
5.5
Advanced non-linear analysis
5.6
Improving price/perf. of Supercomp., Mainframe and
6.1
Sensitivity Analysis
6.2
Enhancing Service and Support
6.4
Computational Procedures
6.4
Grand Average
6.5
Improving price/performance of desktop computing FEA codes
6.7
Training and Documentation
6.9
Design Optimization
6.9
Integration of Design &Analysis
7.5
User Interface
7.7
Automatic Mesh Generation
8.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
Overall — Ratings Are From 10 (Most Important) to 1 (Limited Usefulness)



Linear Static
Piezoelectrics
Electromagnetic Field Analysis
Coupled Heat Transfer/Fluid Flow
Transient Nonlinear Structural
Harmonic Vibration Analysis
Modal Vibration Analysis
Computational Fluid Dynamics
Steady State Nonlinear Heat Transfer
Nonlinear Static Structural Analysis
Fluid/Structure Interaction
Transient Linear Heat Transfer
Bifurcation Buckling
Linear Transient Dynamic
Steady State Linear
Nonlinear Transient Dynamic Structural
Fatigue
0.0%
2.0%
4.0%
6.0%
8.0%
10.0%
12.0%
14.0%
16.0%
% Respondents Planning To Use Who Have Not Already Used

automatic mesh generation. While some these new capabilities primarily benefit the most sophisticated leading-edge users in the short term, vendors hope to drive this new technology into the mainstream, providing a dramatic new set of advanced engineering tools more valuable to analysis specialists and more accessible to designers.

3.3.1 Automatic Mesh Generation

Automatic mesh generation of solid models using tetrahedral elements (TETs) has gained popularity over the past five years. among CAD vendors. Since a tetrahedron is the most fundamental geometric shape, TET elements can be used most easily to fill the broadest range of complex sculptured solid geometry economizing on time for users. Also, TET elements demonstrate less sensitivity to element distortions than hexahedral elements or "brick" elements. Additionally, no viably commercial automatic brick element generator has yet been introduced although substantial progress has been made in recent years.

Tetrahedral elements have never gained the popularity developers hoped. Analysis specialists continue to argue about the relative merits of tetrahedral elements versus hexahedron or "brick" elements. Finite element meshes consisting of brick elements must be generated by the relatively slow process of using mapped mesh procedures. However, users have traditionally employed six sided hexahedral (HEX) elements or collapsed variations such as wedge elements and consequently feel uncomfortable with TETs. Also, the quality of a tetrahedral mesh proves harder to determine.

Also, TETs exhibit artificially high shear stiffness characteristics when used with linear displacement functions. Therefore, analyses involving linear TETs often require unreasonably high element densities to achieve satisfactory results. HEX elements with linear displacement behavior exhibit superior mechanical behavior. Also, most analysts understand HEX element performance better in the presence of material nonlinearities. In fact, analysis guidelines of some large conservative aerospace and defense firms require HEX element use. Proponents of TET elements claim that incorporation of higher order displacement functions provide mechanical behavior that compares favorably with linear HEX elements. Solution times, however, increase substantially. TET element defenders argue that the relative ease of automatic meshing more than compensates for increased solution time. In addition, hardware advances continue to mitigate the TET penalties in solution time.

Boundary element techniques have also received considerable attention for performing analysis on CAD geometry [9]. Advocates claim that boundary element procedures minimize meshing requirements thereby offering the best solution to use CAD geometry for analysis. These procedures either reduce meshing requirements to the surface of a CAD model or eliminate meshing completely. Boundary element techniques promise an obvious time savings in mesh generation. However, their effective use is restricted to bulky solids and lengthy solution times partially offset the benefits.

3.3.2 Design Optimization

Efforts to integrate CAD and analysis along with the availability of constraint based, feature based CAD has spurred the introduction of automated design optimization software. CAD software can define geometric constraints and relationships for the optimization problem. Users define limits on mechanical behavior variables such as deformations, stresses, or natural frequencies. Typical capabilities shipping today demonstrate their greatest capability for shape optimization of individual components or very simple structures subject to static structural loads, steady state thermal loads, and vibration. A complete feedback loop enables optimization routines to update CAD models in the most tightly integrated offerings. Quality of the optimized design depends on the reliability of finite element analyses at each step of the process as well as the robustness and efficiency of the algorithm used to search for the optimum solution. While engineers and designers have been intrigued by such automated design optimization, few success stories have been publicly reported.

3.3.3 Numerical Error Analysis

Error analysis and adaptive finite element analysis have become priorities of the 1990s. The combination partially addresses the issue of error analysis Ñ one of the most elusive questions of finite element analysis.

Error analysis and adaptive finite element procedures represent a feedback mechanism to control the level of numerical error in finite element results. Users benefit from reduction of manpower requirements since adaptivity eliminates the need to the manually generate successive finite element runs. Robust error estimation procedures eliminates numerical or discretization error as a major source of uncertainty .

Four types of adaptive procedures exist Ñ h adaptive, p adaptive, hp adaptive, and r adaptive. H adaptive procedures refer to automatic refinement of a finite element mesh. Software automatically reduces the size of elements and introduces many more of them in parts of the object where software predicts the highest source of error. P adaptive procedures refer to systematic enhancement of the functional behavior of each finite element without refining a mesh. P adaptivity also occurs in regions with the highest estimated error. While h adaptive procedures enrich the reliability of a model by introducing many more elements in trouble spots, p adaptive procedures enrich a model by enriching the predictive power of each element without increasing the their numbers. Hp adaptive procedures, the most comprehensive strategy, incorporates concurrent behavior of h and p adaptive approaches. Hp adaptive procedures address the broadest class of problems in structural analysis. When applied appropriately, they guarantee exponential solution convergence. R adaptive finite element analysis refers to repositioning nodes without increasing the number of finite elements in a model or improving the functional behavior of elements. Nodes are repositioned to bias a finite element mesh towards the regions of the model with the greatest error. R adaptive procedures are usually combined with h and p adaptive procedures.

The effectiveness of each adaptivity approach is problem dependent [6]. P and hp adaptive procedures work best for problems where analysis solutions have smooth functional

behavior except at singular points or inter-element boundaries. The nature of the problem allows those singular points to be represented by nodes. H adaptive solutions are superior for problems where singularities and stress discontinuities cannot be restricted to nodes or inter-element boundaries. Also, each type of adaptivity has different meshing requirements. Commercial finite element software today lacks appreciation of the varying meshing requirements and solution requirements of h adaptive, p adaptive, and hp adaptive capabilities.

Error estimators also need continued development. User feedback indicates that only a select few leading commercial analysis vendors offer reliable h or p adaptive error analysis implementations. Also, no vendor has yet implemented an effective hp adaptive procedure Ñ the most robust and important approach of all.

P adaptive and hp adaptive procedures have the greatest applicability in most design scenarios. A sampling of experienced finite element analysts who use p-adaptive technology report dramatic improvements in solution quality and productivity over the traditional methods of mesh refinement Ñ or h-adaptive approaches Ñ for linear elastic, static, and modal analysis.

3.3.4 Full Physics Simulation

Physical phenomena do not occur independently. Behavior such as heat transfer, structural deformation, fluid flow, and electro magnetism interact. Since design codes allow engineers to consider physical effects independently , simulation experts also simplify their work by analyzing each physical effect independent of the others. However, important problems in many industries cannot be solved unless simulation incorporates the interaction of diverse physical effects . For example, automotive and aerospace engineers must address flow induced vibration. In civil engineering, disasters such as the failure of Washington State's Tacoma Narrows Bridge could have been avoided with a comprehensive evaluation of how wind interacted with the structure. Wind at the right velocity excited a natural frequency of the bridge, causing its collapse.

A new generation of analysis tools under development targets the full range of analyses from linear statics to highly nonlinear "coupled field" problems. These tools remove the barriers between physical processes that are truly coupled but are analyzed today as independent phenomena. Users will also gain other advantagesÑ engineers save time by incorporating a broad range physical phenomena in a single analysis. They also eliminates time spent translating data and modifying models for use with several specialized solvers.

These analysis procedures require substantial computing power that capitalize on new highly efficient finite element solution algorithms [2]. Hardware suppliers are satisfying the need through continuous explosive advances in workstation computing power and the introduction of commercially available massively parallel machines.

3.4 Simulation Challenges

Simulation represents the weakest link in conceptual design for concurrent engineering [23]. Analysis does not keep pace with the rate of design changes due to time requirements for finite element model preparation and concern for reliability of results. Even the leading

automotive companies, deep with expertise, often rely most heavily on physical testing simply because the analysis results arrive too late. The following technology gaps contribute to the shortfall.

3.4.1 CAD Integration

Conflicting design and analysis modeling requirements represent a major barrier in the use of analysis early in the design cycle. Most mainstream finite element preprocessors leverage CAD geometry by allowing analysts to reference the CAD surfaces and then use tools to generate finite element meshes either automatically or manually. Many products also allow users to directly associate a design's operating environmental conditions directly to CAD geometry.

Therefore, if a surface changes during design optimization or the number of elements associated with a surfaces increase during mesh refinement, boundary conditions such as tractions or imposed displacements automatically update.

Despite this progress, a significant technology gap still exists since today's market demands additional functionality to navigate the conflicting requirements between design and analysis. When doing analysis, geometry must be idealized to be consistent with finite element theory and the type of analysis being performed. Detailed three dimensional analysis with solid elements requires direct use of volumes from CAD models. Long skinny members often become 3D "line elements" and thin walled components might be represented by shell surface elements positioned at a midsurface halfway between the top and bottom surfaces of the object. When thin walled objects have complex geometry including characteristics such as double curvature and complex transition regions, extraction of the midsurface can be daunting. Also, shell analyses may require the ability to attach three or more midsurfaces at a common edge. Solid modelers typically allow attachment of only two surfaces at a common edge. The complexity of defining the idealized geometry from CAD becomes compounded when a design includes features such as holes, slots, fillets, chamfers and stiffeners such as ribs, gussets, and face plates. Such features, important for design, are often extraneous in engineering analysis and should be suppressed. Several leading CAD vendors began to support feature suppression and generation of planar midsurfaces.

Recent feedback from users indicates that the capabilities just described do not drive the integration far enough, leaving room for substantial technical innovation. Even with these tools, many simulation specialists still prefer to rebuild their models independently of CAD geometry. They want to generate more computationally efficient models that current integration strategies do not support. Computationally efficient models enable sensitivity analysis and design optimization on a practical timetable. For example, if engineers want to evaluate the load carrying capacity of a long structural members designed as solids, today's automatic mesh generation supports creation of either tetrahedral elements to fill volumes or create shell elements on all surfaces. In many circumstances, neither approach offers a computationally efficient or desired model. Most likely, the simulation expert prefers to represent the members as line elements Ñ a higher level of mathematical abstraction. CAD cannot support such an idealization through automatic means.

The quality of design geometry compounds the problem of automatic generation of simulation models directly from CAD. For example, designers often create geometry with degenerate and overlapping surfaces that might please the eye but are inappropriate for finite element meshing. While CAD vendors and suppliers of analysis software often offer tools to "clean up" the geometry, some geometry manipulation may require the CAD system. Analysis experts are best suited for this work but often they may not have training in CAD. They may have to rely on designers to help them or recreate geometry themselves. Engineers frequently recreate geometry from drawings rather than work with CAD.

3.4.2 Addressing Idealization Errors

While adaptivity eliminates uncertainties due to numerical or discretization errors, the challenge of automatically detecting and responding to idealization errors has yet to be addressed. Engineers must question assumptions while creating their analysis models, even for the most common structural analysis problems. Poor assumptions cause some engineers to encounter difficulties making their analyses converge to a reliable solution with p-adaptive techniques. Consequently, analysts blame the p-adaptive approach and do not suspect an problem defining their analysis model. Such errors are called model idealization errors.. Unlike conventional non-adaptive finite element technology, P-adaptive methods rapidly predict non converging stresses in the idealized corner in an automated way.

Users accustomed to conventional finite element analysis encounter similar problems with point loads, displacement boundary conditions, and interfaces of dissimilar materials when using p-adaptive approaches. Analysts often model applied pressures Ñ i.e., real contact situations Ñ using idealized point loads. The theory of mechanical behavior predicts infinite stresses, or stress singularities, under point loads. P-adaptive error analysis detects these singularities immediately as a nonconverged solution. Mechanical behavior theories also predict singularities at the edge of a region with displacement boundary conditions, and at the interface of dissimilar materials. Therefore, p-adaptive analysis naturally captures and reports errors of idealization which may be overlooked during conventional finite element analysis. Ironically, users become frustrated by the nonconvergence because they are unaware of the implicit assumptions in the idealization of their analysis models, and are surprised by the results.

Therefore, a p-adaptive approach could add value to finite element analysis since it automatically identifies specific finite element idealization errors and reports them with the "nonconverged solution" error message. No software diagnoses unconverged p-adaptive solutions and adjusts for idealization errors.

Beams, plates and shells pose more vexing idealization problems. In reality, beams, plates, and shells are 3D solid objects that exhibit highly complex behavior. Historically, engineers adopted mathematical models that involved simplifying assumptions or "judicious approximations" of mechanical behavior based on the design problem at hand. For example, analysts typically employ three mainstream approaches to evaluate mechanical behavior of a large percentage of shell structures Ñ thin wall shell theory, thick wall shell theory and three dimensional elasticity. Before computers, engineers relied on thin or thick shell theory, being

able to solve only the simplest cases of three dimensional elasticity. Engineers had to judiciously decide which theory would best fit for hand calculations on simplified idealized geometry. In fact, application of thin or thick shell theory often produces unreliable results over much Ñ if not all Ñ of an object's geometry.

Today's finite element practice mimics this philosophy, which predates computers. The established commercial computer codes support thin shell elements, thick shell elements and 3D elasticity elements. Analysts must decide which are best for their applications. Moreover, surveys conducted by D. H. Brown Associates indicate that shell elements are used more than any other 3D element type [26]. While analysts could model shell structures as 3D solids, the size of a valid model would overwhelm most workstations and any PC. Hierarchic analysis of shells, a technique recently developed [1], is being tested to address shell idealization. Hierarchic analysis, in its infancy, needs technical development that extend its capabilities to handle features such as slots and ribs.

3.4.3 Modeling Structural Connectors

The modeling of structural connectors such as fasteners, rivets, welds, etc. represents a significant challenge for simulation software. Additionally, today's practice consumes many hours of preprocessing to generate finite element meshes where nodal locations reflect the exact location of each fastener in a pattern. The complexity of modeling fasteners properly precludes any practical sensitivity analysis or optimization which considers their configuration or spacing as design variables.

3.3.4 Contact and Slippage Simulation

Users also report challenges analyzing contact problems. Although commercial vendors have made progress insuring proper compatibility between two surfaces, users report large differences between finite element results and experimental results. For example, analysts involved in medical applications, report that current commercial capabilities cannot accurately characterize slippage between dissimilar materials such as a metal pin and bone. Therefore, the usefulness of finite element analysis is limited to very gross qualitative understanding of mechanical behavior. Current capabilities cannot replace prototype testing.

3.4.5 Modeling Material Behavior

Simulation cannot be conducted properly without comprehensive modeling of material behavior. Technology to simulate materials must be expanded. As companies move to adopt more advanced materials such as plastics and composites, accurate representation of these materials becomes increasingly critical. For example, automobile companies must comply with increasingly strict environmental standards. A commonly accepted solution involves operation of engines at high temperatures that metals and plastics in today's automobiles could not survive. Therefore, automotive companies will have to incorporate advanced ceramics and composites in future designs.

Material properties have a large affect on results. One example involves simulation to support the design of mold injection equipment. Molten plastic has highly complex behavior. An engineer who benchmarked commercial mold injection simulation software against existing mold injection equipment at his plant reported large differences between the mold injection time predicted by software and the actual time reported at the plant. The user reported that small changes in material coefficients created large changes in numerical results. Therefore, the simulation was unreliable.

4. Data Management

The process of bringing a product to market, from design through manufacturing, poses complex challenges. Today's marketplace demands quick response to changes in technology and consumer demand. Competitive pressures require improvements in quality as well as reductions in manufacturing cost and time to market. Many companies have introduced concurrent engineering Ñ a shift to parallel decision making from the traditional serial process. For example, manufacturing provides feedback on parts during preliminary design. The approach improves designs and reduces time to market. The management challenge, however, becomes more difficult because more people become involved earlier in the design process. Thus, an inherently complex process becomes even more intricate.

The proliferation of engineering software, from 2D drafting on PCs to full-function CAD/CAM/CAE on UNIX workstations compounds data management challenges. While such software products provide users with powerful tools for individual tasks, their output, drowns engineering organizations in a rising flood of documents and data. The benefits of CAD deteriorate when users must spend time finding, moving, and managing data.

Product Data Management (PDM) manages the deluge [18]. PDM becomes an integral part of concurrent engineering, which coordinates expertise up-front at the start of design to avoid extraordinary delays and to drive down the costs associated with change orders and rework late in the development cycle. The focus of a firm's business becomes central to planning an effective PDM solution. Typically, manufacturers target a 50% reduction in time to market along with quantifiable quality and cost improvements.

Product Data Management covers a broad range of applications. Most PDM functions can be grouped into four key areas Ñ document control, engineering process control, configuration management, and system administration.

4.1 Document Control

Many users start with Document Control. Just as library catalog cards record information about books, PDM records refer to electronic parts, assemblies, text files Ñ even paper documents Ñthat constitute the design and manufacturing effort. Document control includes search and retrieval of simple or compound file sets, and managing data compression, storage, distribution and security. Robust products allow users to customize the database to accommodate the different types of documents and information your firm uses, and assign attributes and relationships that tie information together into a cohesive database. Paramount to

users, however, PDM systems must ensure data integrity. The system must provide comprehensive access control Ñ e.g., by project, by group, and by status. The PDM system should ensure that all related drawings and documents have been checked and approved before promoting or releasing an assembly. And while a PDM system should prevent two individuals from unknowingly working on two copies of the same information at the same time, they offer a variety of features for locking, reserving, and sharing information.

"Document centric" companies view the engineering process as culminating in product documentation. Lack of documentation control looms as an acute problem, aggravated by the widespread use of CAD/CAM systems on distributed workstations and PC's. One leading PDM vendor reports over 80% of their customers start implementations with an electronic vault to control files. Electronic or hardcopy documentation forms the basis of communication in most product development processes. Hence, effective management of documentation becomes essential to concurrent engineering efforts.

Most document control efforts start in the CAD area by capturing current document production activity. In this way, firms implement document control on a forward-looking basis. As users demand legacy data, an initiative may be started to capture, or "backfit," old data. Drawings on paper and microfilm may be scanned, cleaned-up, indexed and stored Ñ an expensive and time consuming effort. Since many drawings have little intrinsic value, firms usually opt to capture legacy data on a demand basis, spreading the costs over a lengthy time period.

Firms starting a major new product development are best suited to benefit from a clean-slate document control effort. But customers must assess the risk of coupling a new PDM effort with a new product development program. Properly implemented, however, new projects can benefit from improved document control.

An effective PDM implementation should reflect the role of documentation and product structure from the outset. Strictly vault-focused PDM efforts may ultimately fall short of addressing fundamental problems in the manufacturing business. Many manufacturing firms will eventually require PDM tools that address bill-of-materials and configuration management issues. Those implementing vaults to address short-sighted problems may incur substantial costs to re-architect those systems to satisfy more fundamental needs.

4.2 Process Management

Process Management and Engineering Release Procedures handle dynamic work-in-progress. Process management defines required information, how it moves through the development process, and who has access to it at each stage of the development cycle. Products differ dramatically in their support for process management. Process management for early conceptual design must be flexible, allowing users to share incomplete information among a dynamic group of users. Look for utilities that allow users to distribute, view, edit and annotate information easily. Release procedures define who checks, reviews and approves design information. Supporting these requirements, PDM provides engineering request, order and change control, information organized by project and subprojects, routing and distribution of design information, electronic signatures, and history tracking of who, when and why

various tasks were performed. Sophisticated products support "event actions," that allow users to define automatic steps Ñ e.g., extract reports, run special programs, notify specific users, etc. Ñ the PDM system takes at specific points in the development process depending on pre-defined rules and conditions.

4.3 Configuration Management

PDM developers added release procedures to manage documents as data management software matured, and finally added configuration management and bill of materials to couple MRPII applications to engineering. Configuration Management (CM) enhances the control of design by tracking versions of drawings and other documents for parts and assemblies throughout their life-cycle. CM makes visible relationships between parts in an assembly and how changes are best implemented. PDM functions supporting CM include creating and editing bills of material, CAD integration, version and revision control, report generation and queries. Mechanical design databases are complex, involving hierarchical assemblies with heterogeneous data elements Ñ e.g., models, drawings, specifications, analysis files, change orders, etc. Ñ associated at each level. PDM products differ in their support of complex databases. Look for administrative facilities to construct and edit these relationships, and user facilities to navigate and manage the data they contain. For example, CAD and PDM databases contain overlapping bill of material information. Yet, each application has unique requirements and methods for handling product structure information. Different vendors offer different solutions for managing and reconciling bill of material information from the PDM and CAD environments.

Since products are fundamental to discrete manufacturing businesses, some firms believe that Configuration Management (CM) Ñ defining the content of those products Ñ and not Document Control should dominate the initial PDM effort. A Bill-of-Materials (BOM) represents each product configuration, concisely and formally describing what the company makes. Without accurate BOMs, the firm lacks a clear definition of the product content. Of course firms rely on other devices to support product manufacturing, including the experience of key individuals, drawing packets, expeditors, and a multitude of BOMs supporting different activities. Establishing reliable, explicit product configurations, however, enables a firm to reduce or eliminate numerous non-value-added activities.

Configuration management supports traceability not easily accomplished otherwise. With CM, users can trace a particular part number to a particular version of documentation. For example, a part number identifies a failed part returned from the field. But locating the part's documentation without configuration control is laborious. Another example, world-class companies offering telephone technical support must be able to locate documentation in minutes to answer a query about a particular part or product number. Even product development engineers, wishing to review a previous design for a new application, must be able to locate documentation quickly using part numbers.

4.4 System Administration

Underpinning these application areas are general PDM system requirements. Not all PDM products can handle the same implementation scope. For example, all PDM products provide tools to administer access control. But tools that work well for 20 users on 2 projects may not handle 500 users on 50 projects. Look for customization tools that allow you to modify the user interface, to change or add forms to improve how users interact with the system. A well designed graphical user interface can make the difference between full PDM management of work-in-progress, or a system that never gets used for more than document control. Be cautions, however, that the product architecture allows you to customize the application and remain compatible with future releases. PDM systems by their very nature must integrate with other software applications. Many vendors offer integration tools, services, and pre-integrated applications. As with customization tools, look for integration tools that preserve compatibility with future releases, and support the skills and resources you have available. Some products are customizable with interactive forms, others require sophisticated UNIX script and C programming ability.

5. Gaps in Data Management

5.1 CAD Based PDM

CAD raises unique demands compared to commercial database applications. CAD builds specific data structures to represent geometry. By comparison, non-technical applications manipulate short fields and records Ñ not geometry. The overhead of retrieving data from a relational database management system (RDBMS), while substantially improved over the past few years, remains orders of magnitude greater than direct access with program reads and writes. Storing voluminous CAD geometric data in an RDBMS would raise daunting performance problems. CAD vendors employ fast, proprietary access methods under direct program control. As a result, CAD databases remain "locked-up" Ñ i.e., no common access standards have emerged to date. While CATIA recently incorporated relational databases to store "long-string" tables, the data is only available through the CATIA and RDB combination. Compare this to SQL, the standard for commercial databases that contributed to the creation of powerful and rich development environments supported by CASE and 4GL software tools. STEP/PDES efforts hope to solve the impasse by defining high-level data description standards.

As a reasonable compromise for now, the PDM approach uses the RDBMS to store "metadata," or summary information about the design data. For example, metadata might include part number and revision information appearing in the drawing title block. Even this limited information supports a broad range of applications. The detailed geometric application data, sometimes called "bulk" data, remains outside the RDBMS, leaving each application vendor free to use data structures optimized for their own requirements. With this approach, PDM supports heterogeneous platforms and applications throughout the design cycle across the enterprise, the major strategic focus and advantage of products like Sherpa/PIMS, Hewlett-

Packard's WorkManager, and Metaphase's product data manager framework. Surprisingly, other than Hewlett-Packard, the major systems and database vendors have largely failed to extend their capabilities in the mechanical design vertical market. While IBM and Digital have seriously lagged in meeting mechanical design engineering requirements to date, both have sustained efforts to provide high performance engineering and manufacturing database products.

Another class of PDM products represent extensions of the traditional application suite available from CAD vendors, and offer enhanced functionality based on tight integration with their CAD tools. Vendors like EDS, Applicon and Intergraph now offer integrated assembly modeling. PTC will soon offer in Pro/PDM parameter based dependencies, that alert designers when a dimension or variable that affects their design has been modified. SDRC with the I-DEAS Master series bundles basic PDM functions directly with the modeler for library management and top-down/bottom-up assembly modeling. In general, these offerings focus more on design workgroups.

Users can expect enhanced process modeling and viewing tools, including graphical depictions of engineering processes and product structures. Look for new graphical user interfaces (GUI) to improve the ease-of-use of PDM systems, like those emerging in products from Computervision, Metaphase, and Adra.

To support large implementations, PDM products must address the requirements of dispersed, enterprise-wide operations. Core technologies will be developed to support distributed metadata, and administer large product databases and engineering process models.

Products are becoming so customizable that comparisons of basic features is only a starting point. Visiting reference sites with similar scope and objectives is paramount. Evaluate the vendors track-record for delivering and supporting mature products. Vendors will continue to face conflicting requirements for systems that provide good turnkey functionality to reduce implementation time, and yet support extensive customization to meet individual site's needs.

5.2 Enterprise vs. Workgroup Requirements

The major technology gap in to engineering data management relates to bridging workgroup requirements with enterprise requirements. Firms face different issues summarized by Table 1.1. No data management technology comprehensively addresses all requirements although excellent solutions have been developed specifically for enterprises and workgroups alone.

Firms seeking workgroup solutions address fundamentally different goals than their enterprise advocates. Workgroup solutions address visibility and control of information within the workgroup, e.g., the sharing of data between engineers. Enterprise solutions provide tools to manage the manufacturing business. In particular, they enable the firm to manage major assets more effectively Ñ like inventory and production capacity. They can also enable the firm to manage and meet commitments more effectively, like delivery to customers and subcontractors.

Differences in system size and number of users represents an obvious differentiator between the enterprise and workgroup system. Workgroups generally support fewer than 50

users, whereas enterprise systems often support hundreds or even thousands of users. User and data administration tools must support organizational and project structures at the enterprise level. Otherwise, administration tasks become unmanageable. Access control, for example, becomes complex with matrix responsibilities spanning numerous projects. Simple user and file oriented access control functions overwhelm the administrator charged with maintaining security in a complex environment.

The skill level of the system administrator also varies. Workgroup systems often have one server maintained by an engineer, whereas enterprise systems may have many servers maintained remotely by IT specialists. Specialists have the time and expertise to utilize advanced system administration tools. Workgroup users typically do not.

Product data differs substantially between the enterprise and workgroup level. Enterprise data consists of heterogeneous files originating from diverse groups, e.g., mechanical design, analysis, software, PCB design, etc. Workgroup data originates from a narrow set of applications, resulting in more homogeneity. Also, the scope of workgroup data may be limited to subassemblies and their components, whereas enterprise encompasses the entire product.

Enterprise data also conforms to specific standards, designed to support manufacturing, procurement, and other enterprise functions. As a result, enterprise data has been approved for release with a process that ensures compatibility. But workgroup data reflects the creative development process used to arrive at the "latest" design. While unimportant to the enterprise, workgroups need to manage alternate design concepts, supporting analysis, ancillary programs, and other "design notebook" data that contribute to the final design documents of record.

Table 1.1: Contrasting workgroup and enterprise requirements.

KEY ASPECTS	WORKGROUP	ENTERPRISE
Product structure	Sub-assembly level	Product level
Systems support	Local, user supported	Central/remote, IT function
Number of users	Less than 50	100's-1,000's
Number of files	1K-10K	10's K - millions
Data diversity	Narrow	Broad
Objective	See other engineer's work	Manage operation
Operative goals	Increase individual productivity	Control company's resources, major assets, business commitments
Process orientation	Work breakdown	Engineering change or new product development cycle
Architecture	Client-server	Multi-server
Up-time	Local work schedule	International (24 hour)
Administration	Basic admin tools	Project and organization context sensitive

Formality	Ad-hoc	Controlled
Bulk data	Documents of record and supporting, ancillary, or preliminary data.	Documents of record only.

Fundamental issues must be solved to bridge this gap. For example, workgroup processes are informal, and must be defined by the groups themselves and customized to suite individual project requirements. At the enterprise, or product team level, processes are formal. Both the enterprise process and workgroup subprocesses must work concurrently. The following questions must be answered:

- What kind of data do workgroups release to the enterprise process, i.e., how much information should be visible to the firm? While broad visibility enables people to work and base decisions on information other than the released packet, it increases the potential for error.

- How do the workgroup and the enterprise transfer data? (i.e., how the release and change cycles work?)The procedures must be simple and the tools accessible so as not to interfere with daily work loads.

- When data is placed under workgroup or enterprise control, i.e., when is preliminary data captured by the workgroup system Ñ when is pre-released or partial information captured by the enterprise system? Early capture of conceptual design information, or work-in-progress, enhances visibility and control of the project. It also requires more project discipline and flexible, accommodating tools. Capture of early or partial information by the enterprise system allows manufacturing to start working on long-lead items. But it increases the risk that late changes may result in rework.

When firms have difficulty finding one PDM product that addresses their full spectrum of requirements., they use one of three strategies:

1. Buy a workgroup product and ignore enterprise requirements: Numerous vendors in the PDM and EDMS (electronic data management system) markets offer file management and workflow products designed to support workgroup requirements. These products offer limited support for product structure and change management. Prospects purchase the applications as point solutions addressing relatively narrow data management concerns of a group. Traditional manual systems and/or isolated manufacturing systems manage data outside of the workgroup.

2. Buy an enterprise product and let the workgroups fend for themselves: Some firms

recognize the importance of supporting enterprise-level data and process requirements, but do not provide PDM coordination at the workgroup level. Design information is managed informally within the workgroup until specific, relevant documents are ready for approval and release to the firm. At that time, engineers introduce data to the system for release and access control. Consequently, PDM control begins late in the product development process.

3. Buy an enterprise product to serve both requirements: Some firms extend the role and functionality of an enterprise application to the workgroup. Customers must decide if the tool offers adequate flexibility, accessibility, and cost structure to justify use within the workgroup. Spanning both domains requires considerable attention to the transition required, both in process and data formality.

Marc R. Halpern, Ph.D., P.E, serves as Program Manager, MCAE, CAD/CAM. Dr. Halpern is responsible for research at D. H. Brown Associates covering software technologies and products used for design and analysis. He worked in software development, engineering analysis, and applied mechanics research for over fourteen years before joining DHBA. Dr. Halpern began his career in the early 1980Õs developing nonlinear analysis capabilities for ANSYS from Swanson Analysis Systems Incorporated. His background also includes experience at Weidlinger Associates in software development and engineering analysis, and in the field sales and support organization of SDRC.

Seth B. Hunter, P.E., currently works with Miles Diagmostic Laboratories as a Senior Engineer. Prior to joining Miles, Mr. Hunter was Program Manager, Concurrent Engineering, at D. H. Brown Associates. Mr. Hunter has a wealth of experience applying MCAE/CAD/CAM technology to engineering problems in a broad range of industries. Prior to joining D. H. Brown, Seth served as a Senior Engineer for Pitney Bowes where he developed new methods for dynamic synthesis of high-speed electro-mechanical systems. Previously, Seth supported the mechanical CAD/CAM activities of Philips' affiliates in North America involved in consumer electronics, mold-making and lighting products. He received his Master of Engineering in Mechanical Engineering from Steven's Institute in 1984, and his MBA from NYU in 1991.

References

<u>Periodicals</u>

[1] Babuska, I. , Szabo, B., and Actis, R. Hierarchic Models for Laminated Composites, International Journal for Numerical Methods in Engineering,, 33, 1992, pp. 503-535

[2] Hughes, T. J. R., Franca, L. P., and Mallet, M., A New Finite Element Formulation for Computational Fluid Dynamics, Comp. Meth. Appl. Mech. Eng., 54, pp. 223-234.

[3] Liker J., Fleischer M., Arnsdorf D., Fulfilling the Promises of CAD, Sloan Management Review, Spring, 1992, pp. 74-86.

[4] J. C. Owens, Constraints on Simple Geometry in Two and Three Dimensions, (In preparation)

Books

[5] Geometric and Solid Modeling, An Introduction, Christoph M. Hoffmann, Morgan Kaufmann Publishers, San Mateo, CA, 1989

Proceedings

[6] New Opportunities in Numerical Simulation, Briefing and Workshop, Proceedings,, Engineering Software Research and Development, Donald H. Brown Associates, St. Louis, MO., January, 1993.

[7] C. M. Hoffmann, Modeling the DARPA Engine in Pro/ENGINEER. In DARPA Workshop on Manufacturing, Engineering Design, and Automation, pp. 631-639, Stanford, 1992.

[8] Owen, J. C., "Algebraic solution for Geometry from Dimensional Constraints, "Proceedings of the ACM Symposium Solid Modeling and CAD/CAM Applications, June, 1991, Austin, Texas.

Reports

[9] Casale, M. The Integration of Geometric Modeling and Structural Analysis Using Trimmed Patches, Ph. D. Thesis, University of California, Irvine, 1989
[10]. IBM Clarifies CATIA-CADAM Convergence Plan, D. H. Brown Associates, April, 1994

[11] AutoCAD Designer: The First Days, D. H. Brown Associates, March, 1994

[12] V10 Unigraphics: A New Technology Platform, D. H. Brown Associates, February, 1994

[13] The Status of Solid Modeling: Intergraph's EMS, D. H. Brown Associates, January, 1994

[14] Status of Solid Modeling: Pro/ENGINEER, D. H. Brown Associates, September, 1993

[15] Intergraph at the Crossroads, D. H. Brown Associates, September, 1993

[16] Brown, D. , Engineering Analysis as a Strategic Asset, Soft Prototypes, A Special Report, Penton Publishing, June, 1993.

[17] SDRC's I-DEAS Master Series, D. H. Brown Associates, March, 1993

[18] PIM Leaders, Product Highlights, D. H. Brown Associates, March, 1993

[19] Leaders in Constraint Based Design, D. H. Brown Associates, December, 1992

[20] CV's Focused Solutions, D. H. Brown Associates, July, 1992

[21] SDRC's User Interface: Breaking New Ground, D. H. Brown Associates, May, 1992

[22] 1992 Matra Datavision, D. H. Brown Associates, June, 1992

[23]1992 Conceptual Design Survey, D. H. Brown Associates, March, 1992.

[24]Pro/ENGINEER's User Interface: The Solid Modeling Breakthrough, D. H. Brown Associates, January, 1992

[25]Conceptual Design: Tradeoffs in Performance and Flexibility, D.H. Brown Associates, September, 1991.

[26] 1991 Finite Element Survey, D. H. Brown Associates, May, 1991.

Chapter 6
Summary and Conclusions

Brian K. Seitz
SMS Software Management Incorporated
750 The City Drive South, #420
Orange, California 92668
714/740-1113

Abstract

Over seventy gaps were identified and categorized in terms of domain, function and type. These gaps represent significant opportunities for software developers to develop solutions to enhance the performance of manufacturing enterprise.

Key Words

Attribute; Domain; Gaps; Type

1. Introduction

Within the manufacturing industry, software has held an ever increasing role of importance. Many of the capabilities and efficiencies of today's manufacturing enterprises would not be possible without the information technology of today. However, with that expanded role some issues have surfaced: 1) the time needed to develop and field software which has yielded a gap of desired versus delivered function; 2) the lack of interoperability among software programs; and 3) inflexibility and lack of scaling of programs across different enterprises. Each of these issues has in itself created a series of gaps. The broad domain of manufacturing software holds a wealth of opportunities for the entrepreneur willing to fill these gaps.

As indicated within the approach chapter of this monograph, this document while not an exhaustive study of this problem, is a thorough survey illuminating areas of opportunity where small and large businesses alike may cooperate to fulfill manufacturing needs. The section following will relate findings, developed in previous chapters, from preliminary analysis ton synthesis of the information.

2. Gap Statistics and Demographics

The tables within this chapter provide the reader with an explicit listing of seventy-three specific gaps and gap areas which present opportunities for those engaged in the development, maintenance and operation of software for manufacturing. These gaps represent various perspectives on many of the same issues manufacturing enterprises face today.

The gaps have been analyzed in this monograph in three areas. The first is based on the four domains, which correspond to Chapters 2 through 5 of this document. The second area of analysis was performed based on five identified types. The third was based on four attributes.

A more detailed analysis of these gaps and trends is under development and will be available through a SMS Software Management Incorporated publication during the first quarter 1995. This document can be obtained by contacting the Director of Marketing at 750 The City Drive South, Suite 420, Orange, Ca 92668-4940, 714/740-1113.

2.1 Gap Statistics

The seventy-three gaps span four chapters which view a manufacturing enterprise at various levels of a hierarchy and perspective. Illustrated in Table 6.1 of the gaps identified, approximately 17% were found in the Enterprise Model, 20% in Functional Architecture, 37% in Enabling Integration, and 25% in Design and Data Management domains.

Table 6.1

Gap Domain Statistics Summary					
	Enterprise Model	Functional Architecture	Enabling Integration	Design and Data Mgt.	Total
Percentage	17.8%	20.6%	36.9%	24.7%	100%
Quantity	13	15	27	18	73

The seventy-three gaps when aggregated according to the type of gap, yielded the statistics as illustrated in Table 6.2 below. The results of this clustering indicates there is still a significant amount of software function missing within the present manufacturing environment. The data inferred by this clustering indicates that there is a high degree of overlap between Function, Integration, and Operational Paradigm categories. This implies that the classification scheme of these gap types might be orthogonal to each other. This suggests that a multidimensional mapping of the realm is applicable for a more detailed understanding of the bounds this problem encompasses.

Table 6.2

Gap Type Statistics Summary						
	Function	Integration	Platform	Scalability	Operational Paradigm	Total
Percentage	65.8%	12.3%	2.7%	(2.7%)	19.2%	100 %
Quantity	48	9 (1)	2	(2)[1]	14 (1)	73

The inferences drawn from a second clustering of the data based upon Attributes (Table 6.3) yielded an interesting, though not surprising, finding. As noted in many trade papers today integration of existing capabilities is a significant gap within the industry today, and thus provides a major opportunity for those who develop tools, methods, and techniques to aid or enable this integration.

Though not illustrated within this clustering Graphical User Interface (GUI) was inferred within Functional, Visualization, and Flexibility descriptions. This suggests further study is warranted on the relationships between these categories and human factors.

Table 6.3

Gap Attribute Statistics Summary					
	Semantic Integration	Functional Capability	Visualization	Flexibility	Total
Percentage	45.2%	20.6%	17.8%	16.4%	100%
Quantity	33	15	13	12	73

1 Gap shares two types of classification: either a) Integration and Scalability or b) Operational Paradigm and Scalability

3.2 Gap Demographics

The Gap Type Demographics Table (Table 6.4) illustrates the distribution of gap-types by chapter. This data series yielded two surprising contradictions. First, Enterprise Modeling and Functional Architecture did not have a high correlation between each other. Second, Scalability of solutions did not appear to be of significance in the light of other issues.

It was expected that Enterprise Modeling and Functional Architecture would have a similar set of type demographics instead the reverse was found. This may be due to differing perspectives on how issues were framed; or that Architecture uses models and sees a lack of the ability to change as significant, while Enterprise Modeling develops models and sees the lack of tools to create models as significant.

As expected, there was a high correspondence between Chapter 4. Enabling Integration and Chapter 5. Design and Data Management in regard to the types of issues faced.

Table 6.4

Gap Type Demographics					
	Function	Integration	Platform	Scalability	Operational Paradigm
Chapter 2	8	1	1	-	3
Chapter 3	3	4	1	-	7
Chapter 4	22	2	-	(1)	3
Chapter 5	15	2	-	(1)	1
Totals	48	9	2	(2)[2]	14

The distribution of gaps, depicted in Table 5, illustrates the ability to capture and exchange information, is a recognized gap within the industry and thus presents a significant opportunity for developers with a qualified market. Unlike many new functional capabilities which often have to be proved, integration has an almost innate justification. This obvious justification may not extend to the financial controller's office, however, the industry trends are clear. Currently fielded products lack integration capabilities, this is viewed as a significant gap.

Any products such as:

- which aid or enable integration from analysis tools for capture;
- which help in visualization and understanding of existing data relationships;
- libraries of software objects which specify manufacturing information in a standard exchangeable way, enabling applications to share information.

2 Gap shares two types of classification: either a) Integration and Scalability or b) Operational Paradigm and Scalability

Such products are seen as priority acquisitions today.

Table 6.5

Gap Attribute Demographics				
Chapter	Semantic Integration	Functional Capability	Visualization	Flexibility
Chapter 2	6	3	2	2
Chapter 3	12	-	-	4
Chapter 4	9	8	9	1
Chapter 5	6	4	2	6
Totals	33	15	13	12

3. Gap Detailed Evaluation by Category

Each of the identified gaps have been categorized by Domain, Type and Attribute. These categories are presented in sections 3.1, 3.2 and 3.3 below.

3.1 Gap Detailed Evaluation by Domain

Table 6.6 summarizes the gaps by the primary methods of categorization, i.e. by domain. Because of the significance of the evaluation by domain, further details are provided for each of the four domains.

Table 6.6

GAP Name	GAP Domains			
	Enterprise Model	Functional Architecture	Enabling Integration	Design and Data Mgt.
Application Independence	✓			
Information Collection (adv KE)	✓			
Electronic Manufacturing Glossary	✓			
Dynamic Representation	✓			
Information Sharing	✓			
Semantic resolution	✓			
Natural Language Interface	✓			
CASE Representation Translation	✓			
Business Model Repository	✓			
Presentation Translation	✓			
Reverse Information Engineering	✓			
Information Extraction	✓			
Business function in Software context	✓			
(True) Software Design tools		✓		
Semantic Integration of CAD ad CAPP		✓		

Requirements Definition		√		
Requirements Translation		√		
Market\Production Integration 1		√		
Capabilities based scheduling		√		
Semantic Integration of CAPP and MRPII		√		
Integration Shop Floor Execution and Enterprise Control Systems		√		
Dynamically Reconfigurable Mfg. Control Systems		√		
Standardized Enterprise domain definition		√		
Virtual Enterprise Enablement		√		
Intelligent Data services		√		
Intelligent Products		√		
Horizontal and Vertical Integration Tools		√		
Market\Production Integration 2		√		
Data translators			√	
Requirements Analysis Tools			√	
Semantic Modeling Languages			√	
Schema designers			√	
Data Mapping language or Cross-reference mechanisms			√	
Project Planning			√	
Technology/Resource management			√	
Technology/Resource Modeler			√	
Process Modeler			√	
Data analysis and access tools for existing data repositories			√	
Library of Manufacturing S/W Objects			√	
Interface Code Generators			√	
Technology incorporation aids			√	
Model based operational S/W			√	
Data Browsers			√	
Model-based tools and applications			√	
System Modification and upgrade aids			√	
Data management tools			√	
Decision Support Software			√	
System Performance Monitors			√	
Data Usage Pattern Tools			√	
Information Mgr.			√	
Data traffic Monitors			√	
Library of standard objects			√	
Standard Access Library			√	
S/W Reverse Engineering Tools			√	
Global Standard data access interface			√	

CAD open Architecture				√
PDM Scalability				√
System analysis and Decision Support				√
Adv. Product Data mgt.				√
Design Intent Representation				√
Design Features Library				√
Conceptual Design Capture				√
Design Constraint Representation				√
High level representation of design				√
Development of constraint representation language & Algebra				√
Mesh generation transparency				√
Design optimization				√
Adaptive Design Optimization				√
Full Physics Simulation				√
Complex assembly analysis				√
Contact Slippage Model				√
Material Behavior Modeling				√
Analysis Front End				√

Table 6.7

GAP Domain		
GAP Name	GAP Description	Gap Domain
CAD open Architecture	Creation of modular engineering design software based upon ISO 10303 or some other engineering data integration mechanism	Design and Data Mgt.
PDM Scalability	Ability to scale PDM solution to size and operational modes (formal/informal)	Design and Data Mgt.
System analysis and Decision Support		Design and Data Mgt.
Adv. Product Data mgt.	Ability to perform concurrency/integrity control, check in/out and versioning on product data	Design and Data Mgt.
Design Intent Representation	Ability to unambiguously capture design intent	Design and Data Mgt.
Design Features Library	Ability to create design concepts from parameterized prototypes that include geometry and non-geometry information	Design and Data Mgt.
Conceptual Design Capture	Sketch capture capability and conversion to a tighter geometry specification as design concepts firm	Design and Data Mgt.
Design Constraint Representation	Ability to capture design constraints	Design and Data Mgt.
High level representation of design	see pg 111 DHBrown paper	Design and Data Mgt.
Development of constraint representation language & Algebra	Ability to capture design constraints	Design and Data Mgt.

Mesh generation transparency	ability to generate analysis program input data easily, transparently, or eliminate intermediate data forms for any behavior analysis	Design and Data Mgt.
Design optimization	ability to advise (expert system) on design optimization easily; for any behavior analysis (Struc, Heat, Fluid, etc..)	Design and Data Mgt.
Adaptive Design Optimization	Feedback loop to CAD	Design and Data Mgt.
Full Physics Simulation	Ability to simulate at same time various behaviors (elec, Fluid, Struc, etc..)	Design and Data Mgt.
Complex assembly analysis	ability to easily analyze complex assemblies for physical behaviors (e.g., fastener pattern structural analysis)	Design and Data Mgt.
Contact Slippage Model	Ability to model component interaction more accurately (e.g., predict slippage between two dissimilar materials --bone, stainless steel	Design and Data Mgt.
Material Behavior Modeling	Ability to model dynamic behavior at various materials property states/ranges	Design and Data Mgt.
Analysis Front End	Ability to easily perform physics analysis (FEM, etc.) early in the design cycle	Design and Data Mgt.
Data translators	Data translation/exchange (i.e. import and export)	Enabling Integration
Requirements Analysis Tools		Enabling Integration
Semantic Modeling Languages		Enabling Integration
Schema designers		Enabling Integration
Data Mapping language or Cross-reference mechanisms	Data mapping mechanism (i.e. ability to define synonym relationships between data elements	Enabling Integration
Project Planning		Enabling Integration
Technology/Resource management	Ability to manage resources and technology characteristics data (i.e. library functions)	Enabling Integration
Technology/Resource Modeler	Ability to model resources and technology characteristics	Enabling Integration
Process Modeler	Ability to model process characteristics	Enabling Integration
Data analysis and access tools for existing data repositories		Enabling Integration
Library of Manufacturing S/W Objects	Availability of generic manufacturing s/w objects for sale and reuse within industry. Examples of such are IC catalogs and Standard Part Catalogs.	Enabling Integration
Interface Code Generators	Ability to generate interface or application interconnection code	Enabling Integration

Technology incorporation aids	Tools to aid in the assessment, planning and enablement of new technology assimilation within an manufacturing enterprise.	Enabling Integration
Model based operational S/W	Operational software that incorporates models within the system or is based upon models for behavior.	Enabling Integration
Data Browsers	Ability to access or visualize data elements	Enabling Integration
Model-based tools and applications		Enabling Integration
System Modification and upgrade aids		Enabling Integration
Data management tools	Ability to manage the exchange of data through an infrastructure (i.e. data traffic managers)	Enabling Integration
Decision Support Software	Design and Integration aids and advisors based upon system metrics	Enabling Integration
System Performance Monitors	Ability to monitor performance characteristics and metrics (besides usage and domain)	Enabling Integration
Data Usage Pattern Tools	Ability to monitor, analyze data usage patterns and requirements	Enabling Integration
Information Mgr.		Enabling Integration
Data traffic Monitors	Ability to monitor data access and traffic within an enterprise	Enabling Integration
Library of standard objects	Library of standardized data representations of manufacturing resources	Enabling Integration
Standard Access Library	Library of access/interface software modules	Enabling Integration
S/W Reverse Engineering Tools	Ability to analyze and decode blackbox applications to enable interface/integration within an enterprise.	Enabling Integration
Global Standard data access interface		Enabling Integration
Application Independence	Ability to provide model representation from database representation independence.	Enterprise Model
Information Collection (adv KE)	Tools with the ability to collect business/enterprise information from employees	Enterprise Model
Electronic Manufacturing Glossary	Ability to access common definitions or manufacturing terms in an electronic form.	Enterprise Model
Dynamic Representation	Ability for models to be executed; allowing the dynamic nature of information to be seen	Enterprise Model
Information Sharing	Ability to share captured data and insights within various systems	Enterprise Model
Semantic resolution	Computer-aided thesaurus to assist in the resolution of terms and meanings used within manufacturing enterprise(s).	Enterprise Model

Natural Language Interface	Ability to translate natural language into knowledge-based models	Enterprise Model
CASE Representation Translation	Ability to convert information from one form of Representation format to another.	Enterprise Model
Business Model Repository	Ability to store and retrieve models. A common library of existing models.	Enterprise Model
Presentation Translation	Ability to convert information from one form of Presentation format to another.	Enterprise Model
Reverse Information Engineering	Tools to expose assumptions and rules used to build existing systems.	Enterprise Model
Information Extraction	Tools with the ability to extract useful information from a models.	Enterprise Model
Business function in Software context	Ability to use business analysis techniques and functions within the software development environment.	Enterprise Model
(True) Software Design tools	Ability to link creative thinking tools and techniques with systems development process.	Functional Architecture
Semantic Integration of CAD ad CAPP	Generative process planning and scheduling	Functional Architecture
Requirements Definition	Ability to solicit and encode product requirements from the market.	Functional Architecture
Requirements Translation	Ability to convert operational or functional requirements into structural specifications	Functional Architecture
Market\Production Integration 1	Ability to develop customer requirements in a format for direct input to production planning systems	Functional Architecture
Capabilities based scheduling	Ability to resolve product requirements with existing manufacturing capabilities	Functional Architecture
Semantic Integration of CAPP and MRPII	Ability to drive manufacturing schedules using process orientation as opposed to job/lot	Functional Architecture
Integration Shop Floor Execution and Enterprise Control Systems	Ability to provide linkages to various SFE and Monitoring and Control S/W	Functional Architecture
Dynamically Reconfigurable Mfg. Control Systems	Ability to reconfigure a shop floor control system to run using a new model of operation or product set	Functional Architecture
Standardized Enterprise domain definition	Ability to describe a manufacturing enterprise in a common format to enable various analysis tools to use the same data.	Functional Architecture
Virtual Enterprise Enablement	Ability to connect and exploit a distributed and heterogenous manufacturing system in a transparent manner	Functional Architecture
Intelligent Data services	Ability to distribute complete semantic meaning of data to various application domains and locations	Functional Architecture

Intelligent Products	Tools to enable the ability to create, distribute, and support products with embedded information technology.	Functional Architecture
Horizontal and Vertical Integration Tools	Ability to provide both depth and breath integration; Cross Domain and Scalability	Functional Architecture
Market\Production Integration 2	Ability to develop customer requirements in a format for direct input to material planning systems.	Functional Architecture

3.1.1 Gaps for Modeling of a Manufacturing Enterprise Domain

The Enterprise Modeling chapter provided the reader with a brief overview of modeling. Within Chapter 2 various forms of models, their creation, their usage and inter-relationships were discussed to provide a background for developers to understand the importance and relevance of models to support manufacturing enterprises. Below is a table (Table 6.8) of gaps identified within this chapter.

Table 6.8

Enterprise Modeling Gaps	
GAP Name	**GAP Description**
Information Extraction	Tools with the ability to extract useful information from a models.
Reverse Information Engineering	Tools to expose assumptions and rules used to build existing systems.
Information Collection (adv KE)	Tools with the ability to collect business/enterprise information from employees
Presentation Translation	Ability to convert information from one form of Presentation format to another.
Business function in Software context	Ability to use business analysis techniques and functions within the software development environment.
Business Model Repository	Ability to store and retrieve models. A common library of existing models.
CASE Representation Translation	Ability to convert information from one form of Representation format to another.
Natural Language Interface	Ability to translate natural language into knowledge-based models
(True) Software Design tools	Ability to link creative thinking tools and techniques with systems development process.
Information Sharing	Ability to share captured data and insights within various systems
Dynamic Representation	Ability for models to be executed; allowing the dynamic nature of information to be seen
Electronic Manufacturing Glossary	Ability to access common definitions or manufacturing terms in an electronic form.
Semantic resolution	Computer-aided thesaurus to assist in the resolution of terms and meanings used within manufacturing enterprise(s).
Library of Manufacturing S/W Objects	Availability of generic manufacturing s/w objects for sale and reuse within industry. Examples of such are IC catalogs and Standard Part Catalogs.
Application Independence	Ability to provide model representation from database representation independence.

3.1.2 Gaps for Manufacturing Enterprise Functional Architecture Domain

Chapter 3, Manufacturing Functional Architecture, introduced the reader to the applications and objectives perspective of manufacturing software. Within this section the reader was introduced to what are the important points when discussing a manufacturing enterprise. At this level many operational gaps are seen indicated below in Table 6.9.

Table 6.9

Functional Architecture Gaps	
GAP Name	**GAP Description**
Requirements Definition	Ability to solicit and encode product requirements from the market.
Requirements Translation	Ability to convert operational or functional requirements into structural specifications
Market\Production Integration 1	Ability to develop customer requirements in a format for direct input to production planning systems
Market\Production Integration 2	Ability to develop customer requirements in a format for direct input to material planning systems.
Capabilities based scheduling	Ability to resolve product requirements with existing manufacturing capabilities
Semantic Integration of CAD ad CAPP	Generative process planning and scheduling
Semantic Integration of CAPP and MRPII	Ability to drive manufacturing schedules using process orientation as opposed to job/lot
Integration Shop Floor Execution and Enterprise Control Systems	Ability to provide linkages to various SFE and Monitoring and Control S/W
Dynamically Reconfigurable Mfg. Control Systems	Ability to reconfigure a shop floor control system to run using a new model of operation or product set
Standardized Enterprise domain definition	Ability to describe a manufacturing enterprise in a common format to enable various analysis tools to use the same data.
Virtual Enterprise Enablement	Ability to connect and exploit a distributed and heterogenous manufacturing system in a transparent manner
Intelligent Data services	Ability to distribute complete semantic meaning of data to various application domains and locations
Intelligent Products	Tools to enable the ability to create, distribute, and support products with embedded information technology.
Technology incorporation aids	Tools to aid in the assessment, planning and enablement of new technology assimilation within an manufacturing enterprise.
Horizontal and Vertical Integration Tools	Ability to provide both depth and breath integration; Cross Domain and Scalability

3.1.3 Gaps for Enabling Manufacturing Enterprise Integration Domain

Chapter 4 introduced the concept of integration gaps. It is these gaps which pose the significant challenge of the day which demands further examination. This survey only previewed the depth of the integration issues and suggests a mechanism to identify and

evaluate these gaps. It also listed the tools and mechanisms which are lacking in this field; these tools would enable a manufacturing enterprise to effectively undertake this corrective action.

When identifying gaps the Enabling Integration domain focus yielded a higher quantity of gaps than other perspectives. However, many of these gaps can be viewed as a slightly different flavor of the same root issue. Thus in solving one of these generic problems, it is expected that the solutions will be applicable across a broad range of similar areas.

Table 6.10

Enabling Integration Gaps	
GAP Name	**GAP Description**
Process Modeler	Ability to model process characteristics
Technology/Resource Modeler	Ability to model resources and technology characteristics
Technology/Resource management	Ability to manage resources and technology characteristics data (i.e. library functions)
Project Planning	
Data Browsers	Ability to access or visualize data elements
Data translators	Data translation/exchange (i.e. import and export)
Interface Code Generators	Ability to generate interface or application interconnection code
Data Mapping language or Cross-reference mechanisms	Data mapping mechanism (i.e. ability to define synonym relationships between data elements
Schema designers	
Semantic Modeling Languages	
Requirements Analysis Tools	
Data analysis and access tools for existing data repositories	
Data management tools	Ability to manage the exchange of data through an infrastructure (i.e. data traffic managers)
Global Standard data access interface	
Adv. Product Data mgt.	Ability to perform concurrency/integrity control, check in/out and versioning on product data
S/W Reverse Engineering Tools	Ability to analyze and decode blackbox applications to enable interface/integration within an enterprise.
Standard Access Library	Library of access/interface software modules
Library of standard objects	Library of standardized data representations of manufacturing resources
Data traffic Monitors	Ability to monitor data access and traffic within an enterprise
Information Mgr.	
Data Usage Pattern Tools	Ability to monitor, analyze data usage patterns and requirements
System Performance Monitors	Ability to monitor performance characteristics and metrics (besides usage and domain)
Model based operational S/W	Operational software that incorporates models within the system or is based upon models for behavior.
Decision Support Software	Design and Integration aids and advisors based upon system metrics
System analysis and Decision Support	
System Modification and upgrade aids	
Model-based tools and applications	

3.1.4 Gaps for Manufacturing Enterprise Design and Data Management Domain

Chapter 5 provided the potential software developer with the closest to a detailed trade study of explicit functions lacking in the marketplace today. The Engineering and Data Management domain, while well defined, still has significant gaps in capability. Many such gaps are functional, however, a greater majority of this gaps appear to be within the realm of semantic capture and thus have a high correlation with the previous Enabling Integration chapter. Below (Table 6.11) is an explicit list of the gaps identified within this chapter.

Table 6.11

Manufacturing Enterprise Design and Data Management Gaps	
GAP Name	**GAP Description**
CAD open Architecture	Creation of modular engineering design software based upon ISO 10303 or some other engineering data integration mechanism
PDM Scalability	Ability to scale PDM solution to size and operational modes (formal/informal)
System analysis and Decision Support	
Adv. Product Data mgt.	Ability to perform concurrency/integrity control, check in/out and versioning on product data
Design Intent Representation	Ability to unambiguously capture design intent
Design Features Library	Ability to create design concepts from parameterized prototypes that include geometry and non-geometry information
Conceptual Design Capture	Sketch capture capability and conversion to a tighter geometry specification as design concepts firm
Design Constraint Representation	Ability to capture design constraints
High level representation of design	see pg 111 DHBrown paper
Development of constraint representation language & Algebra	Ability to capture design constraints
Mesh generation transparency	ability to generate analysis program input data easily, transparently, or eliminate intermediate data forms for any behavior analysis
Design optimization	ability to advise (expert system) on design optimization easily; for any behavior analysis (Struc, Heat, Fluid, etc..)
Adaptive Design Optimization	Feedback loop to CAD
Full Physics Simulation	Ability to simulate at same time various behaviors (Elec, Fluid, Struc, etc..)
Complex assembly analysis	ability to easily analyze complex assemblies for physical behaviors (e.g., fastener pattern structural analysis)
Contact Slippage Model	Ability to model component interaction more accurately (e.g., predict slippage between two dissimilar materials --bone, stainless steel
Material Behavior Modeling	Ability to model dynamic behavior at various materials property states/ranges
Analysis Front End	Ability to easily perform physics analysis (FEM, etc.) early in the design cycle

3.2 Gap Evaluation by Type

The gaps are sorted and ordered by type in Table 6.12.

Table 6.12

GAP Type		
GAP Name	GAP Description	Gap Type
Process Modeler	Ability to model process characteristics	Function
S/W Reverse Engineering Tools	Ability to analyze and decode blackbox applications to enable interface/integration within an enterprise.	Function
Adv. Product Data mgt.	Ability to perform concurrency/integrity control, check in/out and versioning on product data	Function
Data management tools	Ability to manage the exchange of data through an infrastructure (i.e. data traffic managers)	Function
Data analysis and access tools for existing data repositories		Function
Requirements Analysis Tools		Function
Semantic Modeling Languages		Function
Schema designers		Function
Data translators	Data translation/exchange (i.e. import and export)	Function
Information Extraction	Tools with the ability to extract useful information from a models.	Function
Standard Access Library	Library of access/interface software modules	Function
Technology/Resource Modeler	Ability to model resources and technology characteristics	Function
Interface Code Generators	Ability to generate interface or application interconnection code	Function
Technology incorporation aids	Tools to aid in the assessment, planning and enablement of new technology assimilation within an manufacturing enterprise.	Function
Intelligent Products	Tools to enable the ability to create, distribute, and support products with embedded information technology.	Function
Intelligent Data services	Ability to distribute complete semantic meaning of data to various application domains and locations	Function
Requirements Definition	Ability to solicit and encode product requirements from the market.	Function
Library of Manufacturing S/W Objects	Availability of generic manufacturing s/w objects for sale and reuse within industry. Examples of such are IC catalogs and Standard Part Catalogs.	Function

Semantic resolution	Computer-aided thesaurus to assist in the resolution of terms and meanings used within manufacturing enterprise(s).	Function
Electronic Manufacturing Glossary	Ability to access common definitions or manufacturing terms in an electronic form.	Function
Dynamic Representation	Ability for models to be executed; allowing the dynamic nature of information to be seen	Function
Natural Language Interface	Ability to translate natural language into knowledge-based models	Function
Presentation Translation	Ability to convert information from one form of Presentation format to another.	Function
Information Collection (adv KE)	Tools with the ability to collect business/enterprise information from employees	Function
Reverse Information Engineering	Tools to expose assumptions and rules used to build existing systems.	Function
Technology/Resource management	Ability to manage resources and technology characteristics data (i.e. library functions)	Function
Complex assembly analysis	ability to easily analyze complex assemblies for physical behaviors (e.g., fastener pattern structural analysis)	Function
Library of standard objects	Library of standardized data representations of manufacturing resources	Function
Data Browsers	Ability to access or visualize data elements	Function
Design Constraint Representation	Ability to capture design constraints	Function
Development of constraint representation language & Algebra	Ability to capture design constraints	Function
Analysis Front End	Ability to easily perform physics analysis (FEM, etc.) early in the design cycle	Function
Design Intent Representation	Ability to unambiguously capture design intent	Function
Full Physics Simulation	Ability to simulate at same time various behaviors (elec, Fluid, Struc, etc..)	Function
Design optimization	ability to advise (expert system) on design optimization easily; for any behavior analysis (Struc, Heat, Fluid, etc..)	Function
Design Features Library	Ability to create design concepts from parameterized prototypes that include geometry and non-geometry information	Function
System Modification and upgrade aids		Function

Contact Slippage Model	Ability to model component interaction more accurately (e.g., predict slippage between two dissimilar materials -- bone, stainless steel	Function
High level representation of design	see pg 111 DHBrown paper	Function
System analysis and Decision Support		Function
Decision Support Software	Design and Integration aids and advisors based upon system metrics	Function
Material Behavior Modeling	Ability to model dynamic behavior at various materials property states/ranges	Function
System Performance Monitors	Ability to monitor performance characteristics and metrics (besides usage and domain)	Function
Data Usage Pattern Tools	Ability to monitor, analyze data usage patterns and requirements	Function
Mesh generation transparency	ability to generate analysis program input data easily, transparently, or eliminate intermediate data forms for any behavior analysis	Function
Information Mgr.		Function
Data traffic Monitors	Ability to monitor data access and traffic within an enterprise	Function
Conceptual Design Capture	Sketch capture capability and conversion to a tighter geometry specification as design concepts firm	Function
Adaptive Design Optimization	Feedback loop to CAD	Integration
Information Sharing	Ability to share captured data and insights within various systems	Integration
Semantic Integration of CAD ad CAPP	Generative process planning and scheduling	Integration
Virtual Enterprise Enablement	Ability to connect and exploit a distributed and heterogenous manufacturing system in a transparent manner	Integration
Horizontal and Vertical Integration Tools	Ability to provide both depth and breath integration; Cross Domain and Scalability	Integration
Integration Shop Floor Execution and Enterprise Control Systems	Ability to provide linkages to various SFE and Monitoring and Control S/W	Integration
Data Mapping language or Cross-reference mechanisms	Data mapping mechanism (i.e. ability to define synonym relationships between data elements	Integration
Global Standard data access interface		Integration
CAD open Architecture	Creation of modular engineering design software based upon ISO 10303 or some other engineering data integration mechanism	Integration
Business Model Repository	Ability to store and retrieve models. A common library of existing models.	Operational Paradigm
(True) Software Design tools	Ability to link creative thinking tools and techniques with systems development process.	Operational Paradigm

Business function in Software context	Ability to use business analysis techniques and functions within the software development environment.	Operational Paradigm
Application Independence	Ability to provide model representation from database representation independence.	Operational Paradigm
Requirements Translation	Ability to convert operational or functional requirements into structural specifications	Operational Paradigm
Market\Production Integration 1	Ability to develop customer requirements in a format for direct input to production planning systems	Operational Paradigm
Market\Production Integration 2	Ability to develop customer requirements in a format for direct input to material planning systems.	Operational Paradigm
Capabilities based scheduling	Ability to resolve product requirements with existing manufacturing capabilities	Operational Paradigm
Dynamically Reconfigurable Mfg. Control Systems	Ability to reconfigure a shop floor control system to run using a new model of operation or product set	Operational Paradigm
Project Planning		Operational Paradigm
Model-based tools and applications		Operational Paradigm
PDM Scalability	Ability to scale PDM solution to size and operational modes (formal/informal)	Operational Paradigm
Model based operational S/W	Operational software that incorporates models within the system or is based upon models for behavior.	Operational Paradigm
Semantic Integration of CAPP and MRPII	Ability to drive manufacturing schedules using process orientation as opposed to job/lot	Operational Paradigm
CASE Representation Translation	Ability to convert information from one form of Representation format to another.	Platform
Standardized Enterprise domain definition	Ability to describe a manufacturing enterprise in a common format to enable various analysis tools to use the same data.	Platform

3.3 Gap Evaluation by Attribute

In Table 6.13 the gaps are sorted and presented by their attributes.

Table 6.13

GAP Attributes	
GAP Name	Gap Element Attributes
Business function in Software context	Flexibility
Design Features Library	Flexibility
Decision Support Software	Flexibility

Semantic Integration of CAPP and MRPII	Flexibility
Application Independence	Flexibility
Analysis Front End	Flexibility
Mesh generation transparency	Flexibility
Dynamically Reconfigurable Mfg. Control Systems	Flexibility
Intelligent Products	Flexibility
Complex assembly analysis	Flexibility
Material Behavior Modeling	Flexibility
PDM Scalability	Flexibility
Interface Code Generators	Function
Schema designers	Function
Data analysis and access tools for existing data repositories	Function
Adv. Product Data mgt.	Function
Semantic Modeling Languages	Function
Data management tools	Function
Project Planning	Function
Technology/Resource Modeler	Function
CASE Representation Translation	Function
Business Model Repository	Function
Information Collection (adv KE)	Function
Adaptive Design Optimization	Function
Process Modeler	Function
Contact Slippage Model	Function
Conceptual Design Capture	Function
Integration Shop Floor Execution and Enterprise Control Systems	Integration (access/semantics)
Virtual Enterprise Enablement	Integration (access/semantics)
Requirements Definition	Integration (access/semantics)
Intelligent Data services	Integration (access/semantics)
Standardized Enterprise domain definition	Integration (access/semantics)
Semantic Integration of CAD ad CAPP	Integration (access/semantics)
Capabilities based scheduling	Integration (access/semantics)
Market\Production Integration 2	Integration (access/semantics)
Requirements Translation	Integration (access/semantics)
Library of Manufacturing S/W Objects	Integration (access/semantics)
Semantic resolution	Integration (access/semantics)
Electronic Manufacturing Glossary	Integration (access/semantics)
Information Sharing	Integration (access/semantics)
(True) Software Design tools	Integration (access/semantics)
Natural Language Interface	Integration (access/semantics)
Reverse Information Engineering	Integration (access/semantics)
Market\Production Integration 1	Integration (access/semantics)
Standard Access Library	Integration (access/semantics)
Design Intent Representation	Integration (access/semantics)
CAD open Architecture	Integration (access/semantics)
Design Constraint Representation	Integration (access/semantics)
High level representation of design	Integration (access/semantics)
Development of constraint representation language & Algebra	Integration (access/semantics)
Model-based tools and applications	Integration (access/semantics)
Library of standard objects	Integration (access/semantics)
Model based operational S/W	Integration (access/semantics)

Global Standard data access interface	Integration (access/semantics)
Full Physics Simulation	Integration (access/semantics)
Data Mapping language or Cross-reference mechanisms	Integration (access/semantics)
Data translators	Integration (access/semantics)
Information Extraction	Integration (access/semantics)
Technology/Resource management	Integration (access/semantics)
Horizontal and Vertical Integration Tools	Integration (access/semantics)
Design optimization	Visualization
Presentation Translation	Visualization
Dynamic Representation	Visualization
System Modification and upgrade aids	Visualization
Data Browsers	Visualization
System Performance Monitors	Visualization
Technology incorporation aids	Visualization
Data Usage Pattern Tools	Visualization
Information Mgr.	Visualization
Data traffic Monitors	Visualization
S/W Reverse Engineering Tools	Visualization
Requirements Analysis Tools	Visualization
System analysis and Decision Support	Visualization

4. Conclusions

Numerous opportunities exist for knowledgeable software developers to work together and with larger manufacturing enterprises to effectively close these gaps.

Significant opportunities exist in areas such as Integration Tools between software systems, Integration Tools on a communications or translation level, and model development and analysis tools which make the understanding the processes and dynamics associated with manufacturing enterprises.

This Monograph is a starting point for the development of improved software for manufacturing and a guide to the existing gaps. Though it is a survey, it is a comprehensive overview of the state-of-the-art in manufacturing software, from the various author's perspectives.

Since software development is in a constant state of flux, this Monograph only reflects on the current challenges of the industry and anticipate the best directions for software development. Thus SMS Software Management Incorporated proposes to update this document in the future to reflect the chapters that are sure to occur.

U.S. Department of Commerce
Technology Administration
National Institute of Standards and Technology
Gaithersburg, Maryland 20899

NIST

OPPORTUNITIES FOR INNOVATION